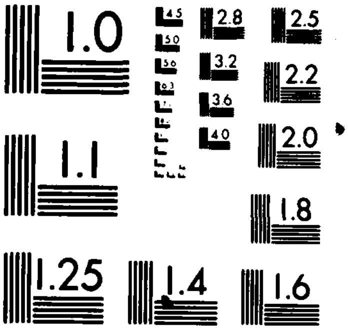
AD-A172 321 AIDING USAF/UPT (UNDERGRADUATE PILOT TRAINING) AIRCREW SCHEDULING USING NETWORK FLOW MODELS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R M HUELSKAMP JUN 86 1/2

UNCLASSIFIED AFIT/CI/NR-86-90T F/G 5/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

0

AD-A172 321

DTIC FILE COPY

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AFIT/CI/NR 86- 90T

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
Aiding USAF/UPT Aircrew Scheduling Using Network Flow Models

5. TYPE OF REPORT & PERIOD COVERED
THESIS/~~DISSERTATION~~

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Robert Michael Huelskamp

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
AFIT STUDENT AT: Massachusettes Institute of Technology

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
AFIT/NR
WPAFB OH 45433-6583

12. REPORT DATE
1986

13. NUMBER OF PAGES
143

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLAS

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Re

18. SUPPLEMENTARY NOTES
APPROVED FOR PUBLIC RELEASE: IAW AFR 190-1

Lynn Wolaver
LYNN E. WOLAVER 6 Aug 86
Dean for Research and
Professional Development
AFIT/NR

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)
ATTACHED.

AIDING USAF/UPT AIRCREW SCHEDULING USING NETWORK FLOW MODELS

by

Robert Michael Huelskamp

B.S., United States Air Force Academy

(1978)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

MASTER OF SCIENCE IN OPERATIONS RESEARCH

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1986

Signature of Author ______

Interdepartmental Program in Operations Research
May 9, 1986

Certified by James B Orlin

James B. Orlin
Thesis Supervisor

Accepted by ______

Professor Jeremy F. Shapiro
Chairman, Interdepartmental Committee on Operations Research

AIDING USAF/UPT AIRCREW SCHEDULING

USING NETWORK FLOW MODELS

by

ROBERT MICHAEL HUELSKAMP

Submitted to the Department of Electrical Engineering and
Computer Science on May 9, 1986 in partial fulfillment of the
requirements for the Degree of Master of Science in
Operations Research

ABSTRACT

thesis

Undergraduate Pilot Training(UPT) is a dynamic, high pressure program which produces front line aviators for the United States Air Force. The eleven month curriculum contains a complicated assortment of academic, flight, and simulator training. Recent expansions in the UPT program and increased emphasis on cost effective scheduling have rendered the current manual scheduling methods obsolete.

This ~~paper~~ develops a micro-computer based scheduling algorithm which aids UPT schedulers in finding feasible solutions to daily scheduling problems. We formulate the problem as a two level network flow problem. Level 1 uses both maximum flow and minimum cost flow formulations to solve the instructor assignment problem. Level 2 addresses student scheduling and uses a two pass, optimization based heuristic which assigns students via a network transportation formulation.

The bi-level network formulation produces feasible daily schedules in minutes on an IBM PC/XT compared to the hours required for the manual method. It also performs feasibility checks at both levels and allows scheduler interaction to effectively generate daily schedules.

Thesis Supervisor: James B. Orlin

Title: Associate Professor of Operations Research and Management

ACKNOWLEDGEMENTS

There are several people I would like to thank for contributing to this thesis effort.

First, I am grateful to the United States Air Force for providing me the opportunity to expand my horizons at such a prestigious institution.

I would like to thank my parents, Bernard and Rita, whose unfailing love and encouragement fostered in me the ability to accomplish my dreams.

I am indebetted to my sister, Maggie, whose fun loving attitude helped me through the rough times and made my stay in Boston most enjoyable.

I am grateful to Vinny Ferrara for showing me that an honest, selfless man can truly excel in our society.

To my good friends at the Operations Research Center especially Carol Holmes, Jackee Kee, and Paul Thompson who served as sounding boards for my ideas and kept me from straying too far afield.

I am especially grateful to my advisor, professor Orlin. His cheerful dispostition and approachable manner was most supportive. More importantly, through his tremendous insight in the area of Network Flows and his ability to clearly communicate complex material, he provided precise guidance and made my most difficult problems more tractable.

Finally, none of this would be possible without the unfailing love and devotion of my wife, Valerie, and my children, Shelly and Steve. Their support and sacrifices provided the strength I needed to start each day, and their welcome arms gave me the comfort I desired at home.

RMH

TABLE OF CONTENTS

ABSTRACT 2

ACKNOWLEDGEMENTS 3

LIST OF FIGURES 7

LIST OF TABLES 8

CHAPTER 1 INTRODUCTION AND BACKGROUND

1.0 Overview 9

1.1 Definition of Terms 11

1.2 Description of Undergraduate Pilot Training

1.21 Curriculum 12

1.22 Organization 13

1.23 T-37 Squadron 13

1.3 Scheduling Environment

1.31 Timeline 16

1.32 Typical Flying Day 17

1.33 Scheduling Process 18

1.34 Recent Development 19

CHAPTER 2 LITERATURE REVIEW

2.0 Introduction

2.01 Problem Description 21

2.02 Problem Constraints 22

2.1 Aircrew Scheduling 23

2.2 Vehicle Scheduling 24

2.3 Vehicle Scheduling Outline 27

CHAPTER 3 INSTRUCTOR SCHEDULING PROBLEM (LEVEL 1)

3.0 Introduction

3.01 Sample Problem 33

3.1 Problem Formulation

3.11 Dilworth Decomposition 37

3.12 Minimal Decomposition Algorithm 38

3.13 Exploiting Special Structure 45

3.2 Maximum Flow Formulation 46

3.3 Minimum Cost Flow Formulation 51

3.4 Heuristic Modifications 55

CHAPTER 4 STUDENT SCHEDULING PROBLEM (LEVEL 2)

4.0 Introduction

4.01 Constraints 60

4.02 Sample Problem 60

4.1 Problem Formulation

4.11 Simple Assignment Problem 63

4.12 Integer Programming 70

4.13 Set Covering/Complete Enumeration 71

4.14 Heuristics 72

4.2 Heuristic Method for the Level 2 Problem

4.21 Step 1 73

4.22 Step 2 74

4.23 Advantages to the Heuristic Method 78

4.24 Problems with the Heuristic Method 79

CHAPTER 5 COMPUTATIONAL RESULTS

5.0 Introduction 85

5.1 Level 1 Program Results 86

5.11 Scheduling Adjustments 91

5.2 Level 2 Program Results 93

CHAPTER 6 CONCLUSIONS AND EXTENSIONS

6.0 Thesis Review 97

6.1 Conclusions 99

6.2 Extensions 100

REFERENCES 102

APPENDICES

A Sample Daily Task Schedules 106

B Sample Unit Contracts 112

C Sample Student Preference Lists 119

D Programs

D.1 Level 1 Program 123

D.2 Level 2 Program 135

LIST OF FIGURES

CHAPTER 1

1.1 Squadron 'Ieierarchy 14

1.2 Unit Makeup 14

1.3 Scheduling Process 19

CHAPTER 2

2.1 Homogeneous, Single-stage Production Scheduling . . . 26

2.2 Non-homogeneous, Single-stage Production Scheduling . . 27

CHAPTER 3

3.1 Network Depiction 35

3.2 Network Depiction of the Dilworth Chains 44

3.3 Maximum Flow Representation 47

3.4 Maximum Flow Solution 50

3.5 Minimum Cost Network Flow Depiction 54

CHAPTER 4

4.1 Sample Assignment Problem 66

4.2 Optimal Solution to the Sample Assignment Problem . . . 68

4.3 Step 2 Sample Assignment Problem 75

4.4 Heuristic Solution to the Sample Assignment Problem. . . 77

4.5 Student Lock-out Infeasibility 81

4.6 Task Lock-out Infeasibility 83

CHAPTER 5

5.1 Phase 2 Schedules for the Week 89

LIST OF TABLES

CHAPTER 3

3.1 Task Schedule 34

3.2 Array Depiction of the Bi-Partite Graph 41

3.3 Labelling Scheme 42

3.4 Conflict Resolution 43

3.5 Staircase Form 45

3.6 Cost Matrix for Feasible Arcs 53

3.7 Instructor Duty Schedule 55

3.8 Least Cost Schedule 56

3.9 Alternative Solutions 57

CHAPTER 4

4.1 Instructor-Task Assignment Summary 61

4.2 Student Preference List 62

CHAPTER 5

5.1 Task Breakdown by Type and Day 87

5.2 Number of Instructors Needed 88

5.3 Unit B Contract, by Instructor 92

5.4 Assigned Schedule of Unit B, by Instructor 93

5.5 Unit B Student Preference Matrix 94

5.6 Level 2 Schedule Output 96

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.0 - OVERVIEW

The United States Air Force's Air Training Command is the largest training operation in the world. Its efforts are diverse as it prepares newly recruited personnel for virtually every USAF role Perhaps the most critical training program within Air Training Command is Undergraduate Pilot Training(UPT). UPT prepares select military officers to be the front line aviators for the USAF. Training is conducted year round at several locations and produces over 2,500 fully qualified pilots annually.

For many years Air Training Command has placed great emphasis on conducting high quality training programs in a cost effective manner. In UPT, the emphasis has been on aggressive use of instructors and aircraft to provide the highest quality pilots. Supporting this effort to better manage resources each flying squadron in UPT has placed increasing importance on daily scheduling efforts. Historically, the daily scheduling has been accomplished heuristically, by hand. Little computer assistance was required or available due to the small size of the training squadrons. However, recent expansions of the UPT squadrons have led to tighter scheduling and increased difficulty in finding feasible solutions to daily training requirements. Paralleling this schedule tightening in UPT has been the introduction of powerful, affordable micro computers. It is my goal to develop a scheduling algorithm on a micro computer which will aid UPT schedulers in finding feasible solutions to daily scheduling problems. In this chapter I will provide a brief background of Undergraduate Pilot Training and a description of the UPT scheduling environment.

1.1 - DEFINITION OF TERMS

DUTY DAY The individual's work day. It begins when the pilot arrives in the squadron and ends when he/she departs.

LAUNCH The initial takeoff of an aircraft mission.

RUNWAY SUPERVISORY UNIT A senior instructor and staff responsible for safe conduct of all air operations.

SLOT TIME The assigned takeoff time(launch time) for an aircraft mission

SORTIE An aircraft mission

SUPERVISOR OF FLYING A senior instructor responsible for monitoring all squadron activities.

TURN TIME The time between successive launches for a particular pilot or aircraft

1.2 - DESCRIPTION OF UNDERGRADUATE PILOT TRAINING

Undergraduate Pilot Training(UPT) is the United States Air Force's school of flying in which all military aviators for the USAF are trained. The purpose of UPT is to provide a high quality, military pilot in a timely manner. The course of training lasts 11 months and exposes the student to all facets of military aviation. The USAF's Air Training Command(ATC) is responsible for the UPT program which is conducted simultaneously at 5 sites throughout the southern United States. The UPT bases are located at Williams AFB., Tempe, Arizona; Reese AFB., Lubbock, Texas; Vance AFB., Enid, Oklahoma; Laughlin AFB., Del Rio, Texas; and Columbus AFB., Columbus, Mississippi.

1.21 - CURRICULUM

The 11 month UPT program is multi-faceted involving several different areas of instruction. The program is divided into three major categories of training. They are 1) Academic(classroom) instruction, 2) Subsonic flight instruction, and 3) Transonic flight instruction. The program begins with one month of basic classroom instruction in aeronautics, aircraft systems, and weather. After this introductory course the students transition into subsonic flight training in the Cessna T-37 jet trainer. This training is conducted by USAF instructor pilots and lasts approximately 16 weeks. During the 16 weeks students receive training in basic aircraft control, takeoff and landing, acrobatics, inclement weather flying, navigation, and two-ship formation. The students "solo" after 15 hours and receive nearly 70 total hours of flight instruction in the T-37.

After mastering the T-37 the student graduates to the Northrop T-38 for additional training in a high speed, transonic flight environment. As in the T-37 the student learns the basic skills needed for acrobatics, navigation, formation, etc. in the high speed environment. T-38 instruction is also provided by USAF instructor pilots and lasts approximately 22 weeks. Simultaneously with flight instruction in the T-37 and T-38 the student continues his academic schedule learning advanced aircraft systems, navigation, and advanced aerodynamics. In addition, the student receives training in various other areas such as ejection, survival, and parachuting.

1.22 - ORGANIZATION

Each of the 5 UPT bases conduct identical training programs throughout the year. The schools operate 50 weeks a year with a new class arriving every 6 weeks. This constant flow of students results in all forms of training being conducted simultaneously at each base. The UPT instructor staff at each base is divided along the same lines as the curriculum. Three squadrons(Academic, T-37, T-38) train independently, and each squadron maintains its own instructor force. The T-37 and T-38 squadrons have similar flight scheduling situations and similar constraints on the program. For this project I will focus on a typical T-37 squadron and its scheduling needs. Only minor constraint modifications would be necessary to apply results to a T-38 squadron.

1.23 - T-37 SQUADRON

The T-37 squadrons at each of the 5 UPT bases are organized in the same fashion. Each squadron operates year round and is divided into 6 training units and a squadron staff. A squadron will have approximately 90 assigned instructors fully qualified to instruct students in all phases of training. The heierarchy in the squadron can be seen in figure 1.1. The instructors in the 6 units are the primary trainers while the other staff instructors divide their time between student training and other duties.

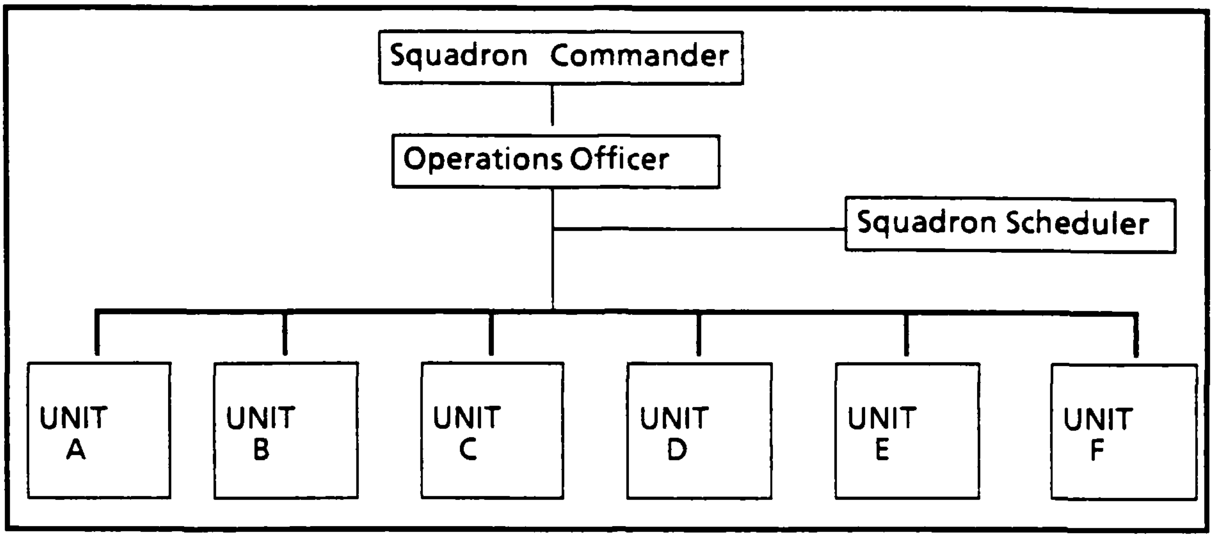


Figure 1.1 - Squadron Heierarchy

The makeup of each unit can be seen in figure 1.2. An average unit consists of a unit commander, his assistant, a scheduler, training officer, and 8 "line" instructors. Although each of these instructors are involved in daily flight instruction of students, the line instructors conduct the bulk of the training due to their lack of additional duties.

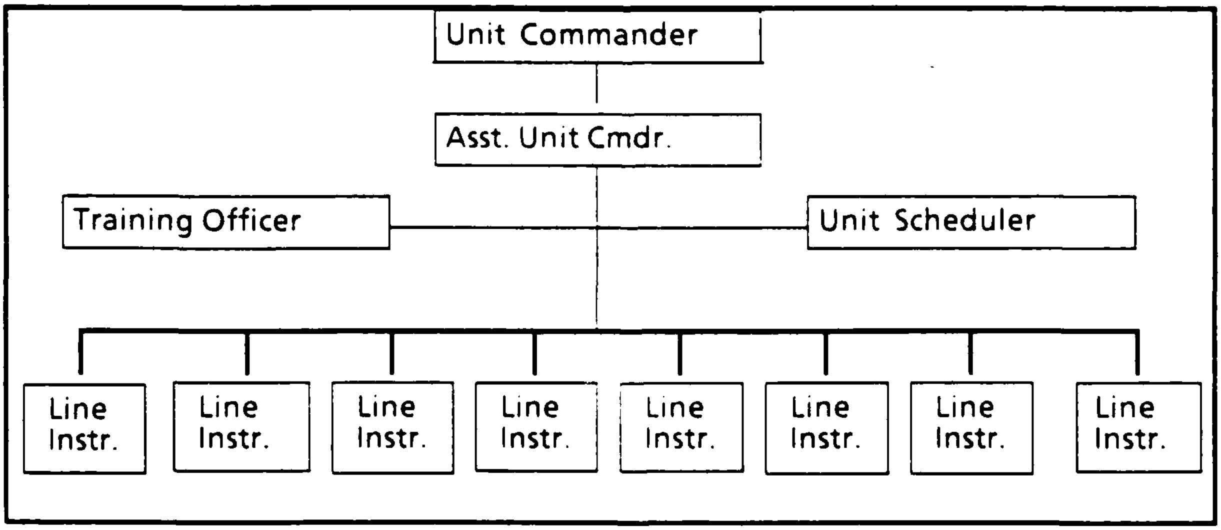


Figure 1.2 - Unit Makeup

Entering classes of students are assigned equitably to each unit as they arrive. Classes average 70 students with a new class entering every 6 weeks. Upon entering, a class is divided between 2 units resulting in a 3:1 student-instructor ratio. Thus a typical T-37 squadron will have nearly 200 students in various stages of training throughout the year.

1.3 - SCHEDULING ENVIRONMENT

Due to the staggered class arrangement in a squadron a wide variety of training is accomplished on a daily basis. This results in a homogeneous training schedule for the entire squadron year round. Each training day has a variety of tasks in the different units, but the distribution of tasks and total missions flown each day is fairly constant barring any unusual delays or losses due to weather, maintenance, etc.

1.31 - TIMELINE

The daily training program is driven by the squadron's timeline. The timeline is a computer monitored training flow schedule which reflects how well the squadron is maintaining its position relative to scheduled graduation dates. The timeline output is broken down into squadron, unit, and individual student levels to exhibit their relative positions. For example, if a student does not fly for an extended period of time due to weather or an illness the timeline will show how many days the student will graduate late if normal training pace is resumed. Due to the various student's flying skills a typical class will have students anywhere from 10 days ahead of the timeline(+10) to 20 days behind(-20). The combination of the student's individual timeline positions reflects the unit's timeline position. Normally, units prefer to remain +2 to +5 days ahead of the timeline to buffer any unexpected losses. Combining the unit timeline positions results in the squadron's position. Typically the squadron's timeline position hovers near 0 as the individual units struggle to maintain their positions.

A second important feature of the timeline output is the correction factors. Each student and unit receive a daily update on the number of sorties needed per day to return to a 0 timeline position by the graduation date. For the student, the printout shows how many missions per day are required to graduate on time. Each flight receives its total missions/day requirement and, finally, the squadron receives input as to the total squadron missions/day needed to smoothly return to a normal training flow.

Allowing an individual student or unit to stray too far from the timeline position can have disasterous affects for the squadron. If too far ahead, students have poor continuity toward the program's end which may lead to unsafe flying conditions. If too far behind, the individual student or unit must put in excessive amounts of overtime resulting in rushed training and fatigue. These are just some of the problems which result from poor timeline management by a squadron.

1.32 - TYPICAL FLYING DAY

The homogeneity of the squadron's flying activities results in a fairly constant squadron-wide daily flying schedule. With the exception of a few night missions, the UPT flight instruction is conducted only in the daylight and twilight hours. The flying day extends from 15 minutes before sunrise to 15 minutes after sunset. Based on the time of year the flying day is anywhere from 11 to 15 hours long.

Agreement with the local FAA Air Traffic Control centers allows a new aircraft sortie to launch from the field every 90 seconds. Due to instructor/aircraft limitations and safety considerations, however, sorties are scheduled to launch with a minimum of 3 minute spacing. This allows for a maximum of 20 sorties per hour from 15 minutes before sunrise to 15 minutes after sunset. These maximums are rarely needed to maintain a good timeline position and primarily reflect a surge capability for the squadron. On average a squadron will need approximately 200 sorties/day to maintain its timeline position and will schedule about 15 sorties/hour.

Although the squadron flies a variety of sorties on a daily basis, nearly every sortie has the same flight profile. Each sortie will depart the home field, last 1.25

hours, and return to the home field for fuel and servicing. Of the total sorties approximately 10% will be flown solo by a student. The other 90% will require one instructor and one student per sortie.

1.33 - SCHEDULING PROCESS

The squadron scheduler is the central scheduler who reports directly to the Operations officer. He is responsible for monitoring squadron and unit timeline positions and allocating daily sorties to each unit based on need and capacity. Unit schedulers have similar monitoring and controlling responsibilities for their respective units and report directly to the unit commander. In all, 10 to 12 schedulers and assistants are actively involved in the process on a given day resulting in nearly 100 manhours daily to ensure a smooth flowing training schedule. The entire scheduling process is accomplished heuristically. The only computer assistance available is the daily timeline output discussed above. In the past year many squadrons have purchased personal computers for the squadron scheduler. However, these are used for little more than as a spreadsheet and as a printing device.

The squadron scheduler is primarily concerned about the total timeline position and is not responsible for the instructor-student pairings for each sortie. To the squadron scheduler all sorties are virtually the same and his primary responsibility is to ensure an equitable and feasible mixture of sorties among the units. The unit scheduler, on the other hand, has a slightly different scheduling problem. He is responsible not only for the number of sorties for his unit, but also for the pilot pairings for each sortie to maintain each student's timeline position.

The scheduling process can be seen in figure 1.3. Two weeks prior the unit schedulers "contract" for sorties based on 1) current timeline position, 2) projected instructor availability, and 3) desired timeline position. Each unit scheduler sends his contract to the squadron scheduler who then determines the necessary daily totals for each day and negotiates with Maintenance Control. An example of a unit contract form can be seen in appendix A. Based on each unit's timeline position and capacity, and the capabilities of Maintenance Control the squadron scheduler attempts to develop a feasible schedule for the entire week. An obvious goal is to simply fill each unit's contract. However, this is rarely feasible due to conflicting contracts from the individual units. The squadron schedule for the entire week is arranged and distributed to the units two workdays prior to the start of the week. Each unit scheduler then manipulates student-instructor pairings on a day to day basis to optimize student training. Appendix B is an example of a weekly flying schedule for the T-37 squadron at Reese AFB.

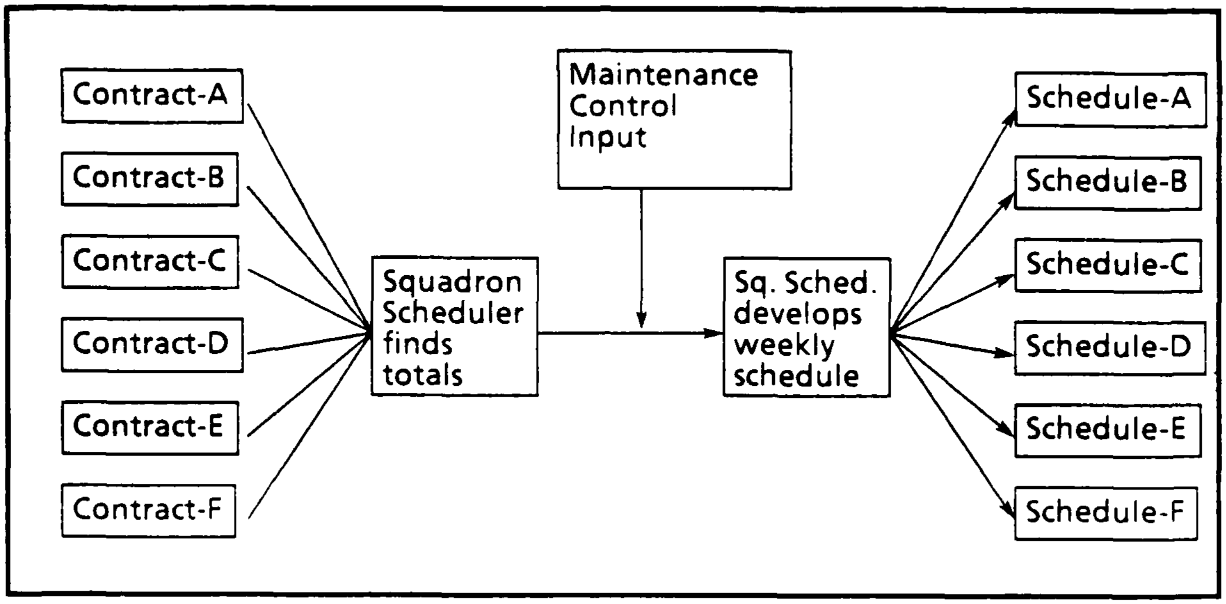


Figure 1.3 - Scheduling Process

1.34 - RECENT DEVELOPMENTS

For several years the method of scheduling currently used by squadron and unit schedulers was adequate due to the nature of training. Smaller class sizes and more dependable aircraft systems resulted in fewer timeline constraints. This made squadron and flight scheduling easier as feasible solutions were more readily available. Typically an average duty day for an instructor was 8-10 hours and weekend flying was unheard of. Suboptimal feasible solutions from the units easily met timeline constraints and the squadron scheduler merely filled each unit's contract. No effort was made to maximize training as this would only result in early graduation and subsequent sequencing problems with other training programs.

Since 1980, however, the need for optimal scheduling has become paramount. Class sizes have increased significantly and constraints on the system have become much more binding. Feasible solutions to daily timeline scheduling requirements are much more elusive. A common approach used by many squadrons to find feasible solutions has been to increase the instructor duty day and to include weekend flying. 6 day work weeks are now the norm at most UPT bases. The manual heuristic method is becoming less effective and squadron and unit schedulers have less time to devote to scheduling since they, too, must fly more to meet timeline demands. All of these developments have significant morale, training, and safety implications as instructors and students work longer hours and become more fatigued.

CHAPTER 2

LITERATURE REVIEW

2.0 - INTRODUCTION

Under the current scheduling system it is virtually impossible to determine an optimal schedule due to the difficulty of the problem and time constraints. In the past no real attempt was made at computer aided scheduling due to the lack of facilities. The size of the training squadron was not large enough to justify powerful and expensive computer equipment used only for daily scheduling and data management. However, with the advent of the personal computer sufficient power may now be available to efficiently find nearly optimal solutions to the UPT scheduling problem. In this chapter I will further define the UPT scheduling problem by addressing the problem description, constraints, and structure.

2.01 - PROBLEM DESCRIPTION

The UPT scheduling problem can be subdivided into three basic levels. The first and most constrained level is the squadron scheduler's task of assigning instructor pilots to the daily task schedule(Level 1). Recall that each task has a fixed start time and each instructor has limited availability. After determining a feasible instructor-sortie matching for the entire squadron, the schedule is divided into the six independent flying units. This second level of scheduling is performed by the unit scheduler who attempts to find a feasible student-instructor matching for all the unit's assigned tasks. The third level of the problem is addressed after the unit scheduler "fills" the schedule with feasible student/instructor matchings. At this level the unit scheduler determines the mission category to be flown by each student.

The first two levels of the problem are actual scheduling problems and are similar in many respects. Their specific constraints and structure will be addressed in the next two chapters. The third level, mission category assignments, is a data management problem in which the unit scheduler develops a priority queue for each student. A First-In/First-Out queue of mission categories ensures a balanced mix of mission types and good continuity of training.

2.02 - PROBLEM CONSTRAINTS

The squadron and unit schedulers work in unison to to find a feasible scheduling solution which maintains a smooth training program that meets Air Training Command guidelines for student training. Of the myriad of constraints on the system the following are most binding for the schedulers.

Level 1

1. 3 tasks maximum per day for each instructor pilot
2. 2.75 hour turn time for each instructor
3. Unit report time for all instructors is 1.75 hours prior to the 1st unit sortie of the day
4. Maximum flying day is 12 hours for each instructor
5. Maximum duty day (flying – additional duties) is 16 hours

Level 2

1. 3 tasks maximum per day for each student pilot
2. 2.75 hour turn time for each student
3. Unit report time for all students is 1.50 hours prior to the 1st unit sortie of the day
4. Student duty day must be less than 12 hours
5. Each student flies with a maximum of 5 different instructors during training

These and other training/scheduling constraints are contained in Air Training Command manual 51-37. In addit'on to these "hard" constraints several "soft" constraints exists to provide optimal training. If possible, these constraints should be met as well.

1. Limit instructor duty day to 12 hours
2. Minimum of 1, maximum of 2 sorties per student per day
3. Maximum of 2 sorties per instructor per day
4. One sortie per day for staff/support personnel
5. Minimize instructor idle time between sorties
6. Five day work week

Obviously as each constraint is considered the difficulty of obtaining a feasible solution increases. Some constraints are more binding than others and dramatically increase the difficulty of the scheduling problem.

2.1 - AIRCREW SCHEDULING

The area of aircrew scheduling is one that has recieved much attention in recent years. In aircrew scheduling the primary concern is to sequence the movement of crew members in space and time to staff desired airplane movements. Usually the aircrew schedule is determined after the airplane schedule since the airplane costs significantly dominate the crew costs.

Some of the best results in crew scheduling in general have been obtained in those areas where the problem can be formulated as a network flow model[43]. Unfortunately for the airline industry the variety of airplanes, crew domiciles, and complicated pilot workrules do not lend themselves to easy network formulation. The airline crew scheduling problem is generally broken down into two parts - generating pairings and constructing bid lines. A pairing is a sequence of trips that begin and end at the same domicile which one crew must complete. A bid line is a set of pairings that represent a monthly work schedule for a crew.

Baker[1] uses a set of heuristics to solve the pairing problem. He has solved a single domicile/900 trip crew scheduling problem for Federal Express Corporation to within 1% of the best known result. For bid line construction Finnegan[23] uses a matching algorithm to develop bid lines given a set of pairings. Marsten and Shepardson[46] use set patitioning ideas to solve both the pairing and bid line problems.

2.2 - VEHICLE SCHEDULING

Although the UPT crew scheduling problem involves aircraft and aircrew members, its network structure is much simpler when compared to the scheduling in the airline industry. This structure reduces the UPT problem from a complex airline scheduling problem to one which can be modelled as a production planning or vehicle scheduling problem. Vehicle scheduling problems can be viewed as routing problems with additonal time constraints imposed on the network. As a result, the tasks are accomplished in both space and time. Vehicle scheduling problems can be broken down into three major classes -- arc-based scheduling, node-based scheduling, and a combination of both. In the arc-based scheduling a set of arcs in the network must be traversed. Examples include snow removal and the Chinese Postman problem with time constraints. In the node-based scheduling a collection of supply and demand nodes are given and at least one vehicle must travel from each supply node to its corresponding demand sight. Examples of this type of network include the dial-a-ride problem and the assignment problem. Finally, in the combined problem there specific nodes and arcs which must be serviced. School bus scheduling, where specific schools and streets must be serviced within time constraints, is an example of this type of scheduling[180]

In simple terms, the vehicle scheduling problem involves vehicles, customers, and a time schedule. The vehicles travel from customer to customer servicing customer needs(making deliveries, pickups, etc.) at specific times. The goal of the scheduler is to "optimally" route each vehicle such that all customers are serviced on time. In production scheduling, the "vehicles" are actually machines on which certain jobs(customers) must be completed. In this context the goal is to optimally schedule jobs to machines such that all jobs are completed on time.

Translated into the terms we are using for the UPT crew scheduling problem, the instructor pilots(level 1) and students(level 2) can be viewed as the "vehicles" or "machines" required to perform specific jobs. The jobs are tasks which must be accomplished at predetermined times. The scheduler's goal is to optimally assign tasks(jobs) to pilots(machines) such that all tasks are accomplished on time. Figure 2.1 is an illustration of the level 1 problem(matching instructors and tasks) viewed as a production scheduling problem.

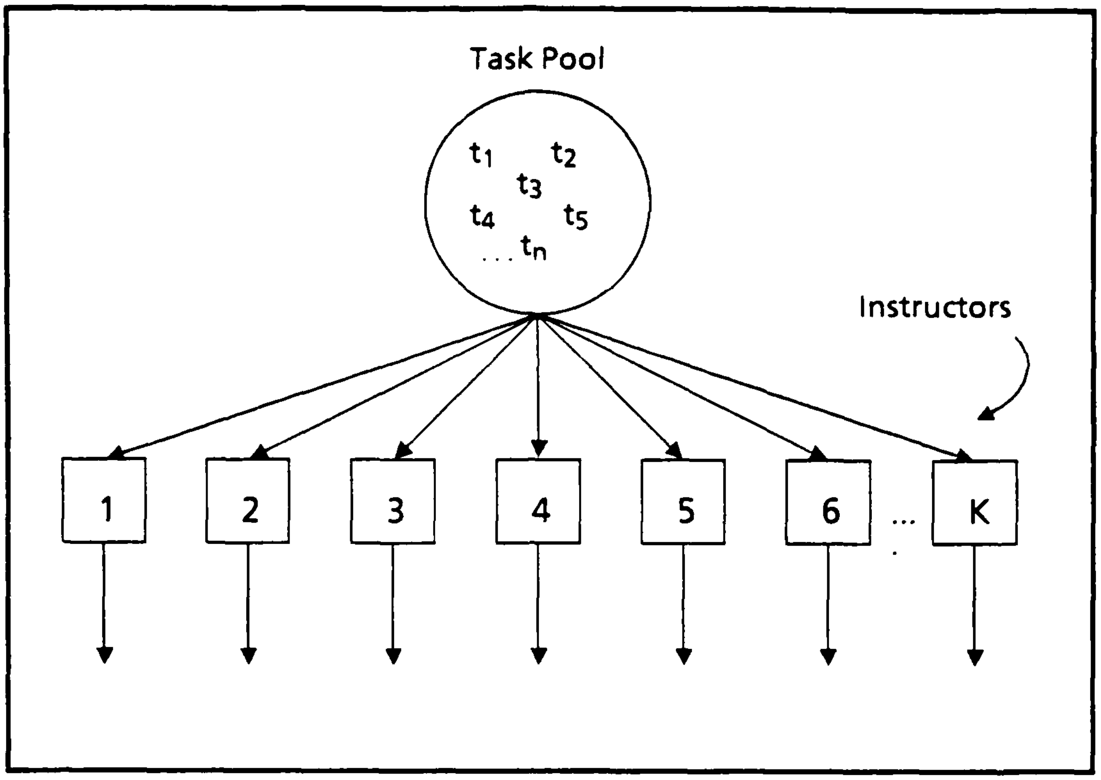


Figure 2.1 - Homogeneous, single-stage production scheduling

Our goal is to sequence the daily tasks $t_1,t_2,\ldots,t_k$ through the instructors in an "optimal" manner subject to the constraints outlined in the previous section. Each task has a predetermined start/completion time and each pilot has a constraint of accomplishing only one task at a time. What an "optimal" matching of tasks to pilots actually is will be addressed in the next chapter. Figure 2.2 shows the level 2 problem which incorporates not only the time element but also the matching element. The task type must match with the student since the students are non-homogeneous.

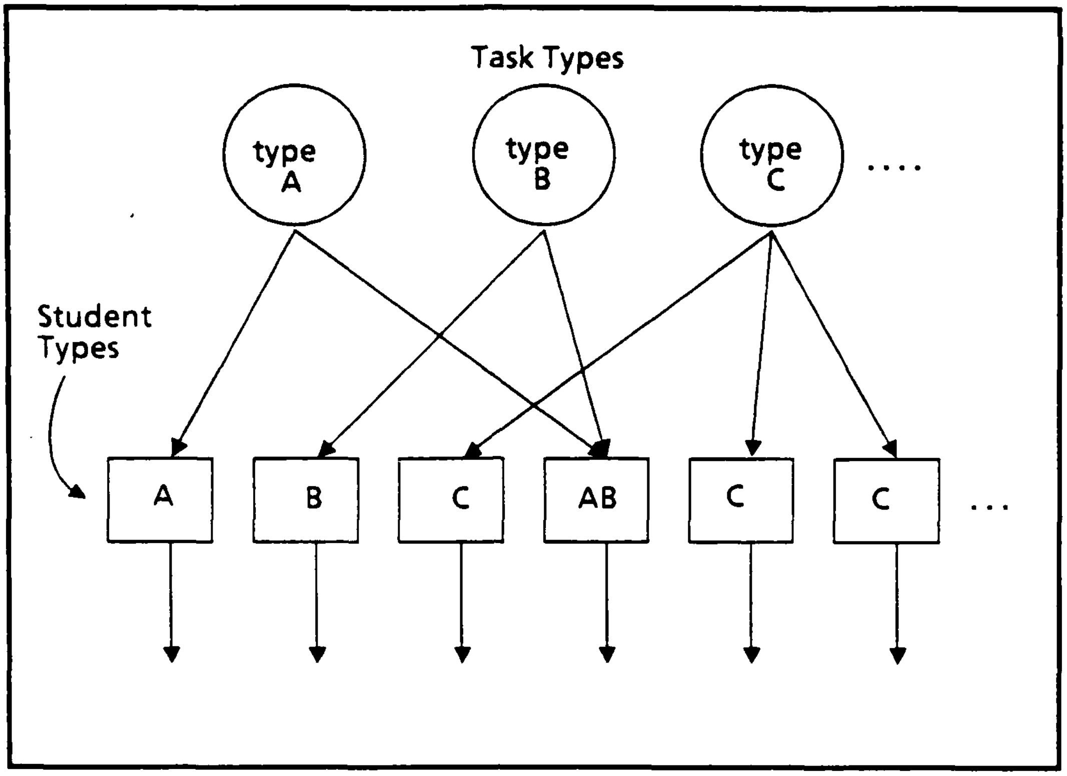


Figure 2.2 - Non-homogeneous, single-stage production scheduling

2.3 - VEHICLE SCHEDULING OUTLINE

The following outline from Bodin and Goldin[12] outlines a number of characteristics that further describe any vehicle scheduling problem. One can see from this outline that the UPT scheduling problem neatly fits into the category of pure vehicle scheduling problems. I have asterisked the item in each category which applies to the UPT problem.

A. Time to service a particular node or arc
   1. time specified and fixed in advance (pure vehicle scheduling problem) (*)
   2. time windows (combined vehicle routing and scheduling problem)
   3. time unspecified (vehicle routing unless precedence relationships exist)

B. Number of domiciles
  1. one domicile (*)
  2. multiple domiciles

C. Size of vehicle fleet available
  1. one vehicle
  2. multiple vehicles (*)

D. Type of fleet available
  1. homogeneous case (all vehicles the same) (*)
  2. heterogeneous case (all vehicles not the same) (*)

E. Nature of demands
  1. deterministic (*)
  2. stochastic

F. Location of demands
  1. at nodes (not necessarily all) (*)
  2. on arcs (not necessarily all)
  3. mixed

G. Underlying network
  1. undirected
  2. directed (*)
  3. mixed

H. Vehicle capacity constraints
  1. imposed-all the same (*)
  2. imposed-not all the same
  3. not imposed

I. Maximum vehicle route-times
  1. imposed-all the same (*)
  2. imposed-not all the same
  3. not imposed

J. Costs
  1. variable or routing costs
  2. fixed operating or capital costs (*)

K. Objective
  1. minimize routing costs incurred
  2. minimize sum of fixed and variable costs
  3. minimize number of vehicles required (*)

L. Other (problem dependent) constraints

In category A the time to service a particular item is time specified and fixed in advance. As was mentioned before, this applies to the duration of a task(sortie length). This length is directed by regulation to be 1.25 hours for aircraft and simulator sorties. This item also applies to the specified start/finish time for each task. Again, the start time for each sortie is fixed in advance by Maintenance Control. Limited flexibility is available with respect to start times, but this is utilized only if no feasible schedule can be determined. The primary reason for strict start times is that several sorties are flown in a single aircraft or simulator. Obviously, if a sortie starts late all subsequent sorties in that aircraft/simulator must be pushed back proportionately. This rippling through the schedule may result in an infeasible turn time for a pilot flying that aircraft later in the schedule.

In category B the number of domiciles is limited to one. All sorties originate and terminate at the home airfield. This simplifies the problem since it allows greater flexibility of sortie assignments to pilots.

Category C, the size of the vehicle fleet, further defines the problem. The UPT problem has more than one vehicle(pilot) available for assignment to the multitude of tasks.

The type of fleet available, Category D, involves a set of homogeneous instructor pilots all possessing the same flying qualifications for the level 1 problem. The level 2 problem involves a set of non-homogeneous students which must be matched to instructors.

Categories E and F address the nature of demand at each node. The nodes in the UPT problem are the sorties to be flown. The demand at each node is deterministic. That is, each task "demands" exactly one instructor(level 1) and one student(level 2).

The underlying network, Category G, in the UPT problem is directed. This is so because of the precedence relationships among tasks. For example, of the five sorties assigned to a particular aircraft($s_1, ..., s_5$) each has a specified start time. The sequence must be flown in ascending numerical order resulting in the directedness of the network.

In Category H, capacity constraints are imposed and are not necessarily the same for each pilot. The constraints in the previous section indicate that the maximum number of sorties per day is three for each pilot. Particular pilots (Commanders, staff, etc.) may be limited to a maximum of two or less due to additional duties. Unfortunately, this restriction on capacity for each pilot has been proven to be NP-hard [4]. Methods to accomodate this difficulty will be discussed in the following chapter.

Category I involves maximum vehicle route times. This category captures another major constraint of the UPT scheduling problem. Maximum route times equate to the maximum flying duty day for the pilots. As with the capacity constraints in Category H the duty day restriction adds considerably to the difficulty of the problem.

Category J- costs. The costs in the UPT scheduling problem can be viewed as fixed. There are no variable costs associated with overtime, instructor duty days, etc. The capital costs associated with pilot acquisition is a long term planning concern not a daily scheduling matter.

The primary objective of the UPT scheduler, Category K, is to minimize the number of pilots needed to fill the daily schedule. This objective can be used to determine if a particular task schedule is infeasible(i.e. is the minimum number of pilots needed greater than the number available?) It also indicates the number of pilots available for additional duties. Thus, the "minimize the number of pilots" objective allows the scheduler to iterate through feasible schedules and to perform sensitivity analysis on the chosen schedule.

Finally, category L captures any remaining problem-dependent constraints. In our case this would include additional constraints such as unit report times which affect the length of a pilot's duty day.

In summary, The problem structure of the UPT scheduling problem can be viewed as a pure Vehicle Scheduling problem with a single domicile containing multiple homogeneous (level 1) /non-homogeneous(level 2) pilots. The task demands are deterministic at each node in a directed network of tasks. Pilot capacity and time constraints are imposed and the objective is to minimize(Level 1)/maximize(Level 2) the number of pilots needed to fill the schedule.

In the following two chapters I will address each level of the UPT scheduling problem in turn. Chapter 3 will focus on the formulation of the instructor scheduling problem while chapter 4 adresses the student - instructor matching problem.

CHAPTER 3

INSTRUCTOR SCHEDULING PROBLEM (LEVEL 1)

3.0 - INTRODUCTION

In the formulation of the ATC Crew Scheduling problem I will divide the problem into three separate areas. The first area to be addressed is determining the minimum number of instructors needed to fill the daily schedule(Level 1). This is by far the most binding constraint on the problem as a whole since the instructor force is small compared to the number of students. As was mentioned in chapter 1 the instructor scheduling problem is addressed on a daily basis by the squadron scheduler. The squadron scheduluer is responsible for scheduling all of the instructors equitably and within Air Training Command regulations and squadron guidelines[57]. In chapter 2 I determined that these regulations and guidelines coupled with the characteristics of the squadron indicate the problem is a pure vehicle scheduling type. The problem involves a single domicile containing multiple homogeneous pilots. The task demands are deterministic at each node in a directed network. Pilot capacity and time constraints exist, and the primary objective is to minimize the number of pilots needed to fill the daily schedule.

In formulating the Level 1 problem I will first look at a relaxed version without capacity and time constraints since these constraints render the problem NP complete. I will then look into methods to accomodate these restrictions.

3.01 - SAMPLE PROBLEM

To aid in visualizing the formulation of the problem I will use the following sample problem. The ten tasks listed in table 3.1 represent a small portion of a realistic schedule encountered by a squadron scheduler on a daily basis. Actual problems may have well over 200 such tasks to schedule daily (see appendix A).

The table includes a brief description of each task, the amount of pre-briefing time required, the task start time, the duration of the task, the amount of de-briefing required, and the total time period needed to accomplish the task. Each task is accomplished by a single instructor.

The sample problem can also be viewed as a network flow problem (figure 3.1). In this depiction the nodes are the tasks t_1, $t_2, \ldots, t_{10}$ and the directed arcs represent feasible paths through the nodes. The time block required for each task is included in the node. An arc exists from nodes t_i to node t_j if the start of the time block for t_j is later than the end of the time block for t_i. Some arcs are not included in the network to prevent the depiction from becoming too cluttered. Obviously, if arcs exists from node t_i to node t_j and from node t_j to node t_k then an arc also exists from t_i to t_k.

Table 3.1 - Task Schedule

Task No.	Task Description	Pre-Brief Time B_i	Task Time		De-brief Time DB_i	Time Block
			Start S_i	End E_i		
1	Runway Supervisor(RSU)	15	0545	1100	15	0530-1115
2	Supervisor of Flying(SOF)	15	0615	1035	15	0600-1050
3	aircraft mission	60	0615	0735	40	0515-0815
4	aircraft mission	60	1015	1135	40	0915-1215
5	aircraft mission	60	1200	1320	40	1100-1400
6	simulator mission	30	1200	1315	30	1130-1345
7	aircraft mission	60	1450	1610	40	1350-1650
8	simulator mission	30	1430	1545	30	1400-1615
9	simulator mission	30	1350	1505	30	1320-1535
10	simulator mission	30	1700	1815	30	1630-1845

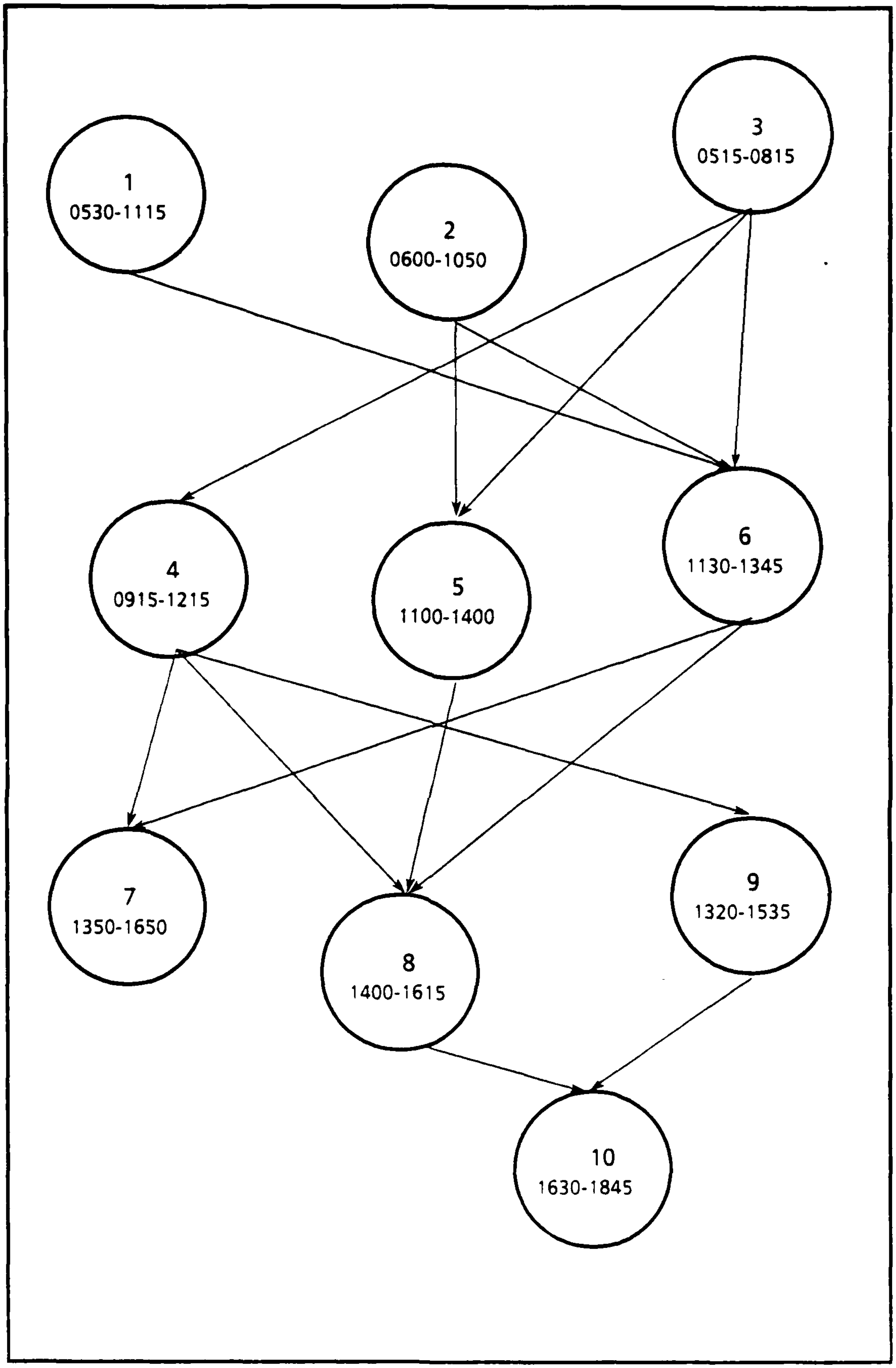


Figure 3.1 - Network depiction

3.1 - LEVEL 1 PROBLEM FORMULATION

The initial goal of the squadron scheduler is to merely find a feasible schedule of instructors which covers all the tasks. By determining the minimum number of instructors required to accomplish all assigned tasks the scheduler can quickly check for infeasibility. That is, if the minimum number of instructors needed is greater than the number available no feasible solution exists. If a feasible solution exists the scheduler can add optional, low priority tasks such as meetings, training sessions, etc. to any extra instructors to use the instructor pool to a maximum extent. In addition to determining the minimum number of instructors needed to fill the schedule the scheduler can attempt to minimize some sort of cost function as well. Costs can be assigned to indicate the desirability of a particular pilot's schedule. The cost is directly proportional to idle time in the schedule. Accomplishing all the tasks with little or no idle time between tasks is much more desirable than having large amounts of idle time in the schedule. Using the minimum number of pilots as an input to a cost(idle time) minimization problem the scheduler can find a low cost feasible solution to the Level 1 problem.

Using the directed, acyclic network representation in figure 3.1 the problem reduces to finding the minimum number of disjoint paths which cover all the nodes in the network. Once these paths have been found, each path can be interpreted as a schedule for a particular instructor. The squadron scheduler can then assign each available instructor a set of tasks corresponding to a path in the network. This problem, which is often referred to as a Dilworth decomposition problem.

3.11 - DILWORTH DECOMPOSITION

Ford and Fulkerson[28] give an excellent description of how to solve the Dilworth decomposition problem using a partially ordered set. A partially ordered set P is a set with a transitive, antisymmetric order relation "$\rangle$" where $t_i \rangle t_j$ is represented in the network by a directed arc from node t_i to t_j . The Level 1 problem can be written as a partially ordered set in the following manner.

(3.1) $t_i \rangle t_j$ if and only if $S_j - B_j \leqq S_i + D_i + DB_i$.

That is, task t_i must be completed prior to the pre-briefing time for t_j. Figure 3.2 is a partially ordered set. A *chain* in P is a set where

(3.2) $t_1 \rangle t_2 \rangle . . . \rangle t_k$.

This coincides with a directed chain in the network. A *decomposition* of P is a partition of the partially ordered set into disjoint chains. This is also referred to as a chain cover. The decomposition is *minimal* if it covers the nodes with the smallest number of chains. Finally, two distinct members of P are *unrelated* if neither $t_i \rangle t_j$ nor $t_j \rangle t_i$. These unrelated members form the antichain.

> **Dilworth's Theorem**: *Let P be a finite or countably infinite partially ordered set. then the cardinality of the minimal decomposition (chain cover) is equal to the cardinality of the maximal number of unrelated members (antichain) of P.*

By connecting Dilworth's theorem to the following Max Flow theorem we can solve the problem of finding the minimal chain decomposition[28].

> **Konig-Egervary Theorem**: *Let G = [S,T;A] be a bipartite graph. The maximal number of arcs of G that are pairwise node disjoint is equal to the minimal number of nodes in a S, T disconnecting set of nodes.*

The disconnecting set of nodes is a set which blocks all chains from S to T.

According to Ford and Fulkerson

> Another statement of this theorem is sometimes given in terms of *m* by *n* arrays that contain two kinds of cells, admissible and inadmissible. Suppose we refer to the rows and columns of the array by the common term "lines". A set of lines *covers* the admissible cells of the array if each admissible cell belongs to some line of the set. A set of admissible cells is *independent* if no two cells of the set lie in the same line. By constructing from the array the bipartite graph G composed of nodes
>
> (3.3) $S = \{x_1, \ldots, x_m\}, \quad T = \{y_1, \ldots, y_n\}$
>
> and arcs (x_i, y_j) corresponding to admissible cells, one sees the notion of "independent set of admissible cells corresponds to "pairwise node disjoint arcs", and hence the Konig-Egervary theorem becomes: *the maximal number of independent admissible cells is equal to the minimal number of lines that cover all admissible cells.*

For the Level 1 problem, then, the maximal number of independent admissible cells is equal to the minimal number of instructors to accomplish all the tasks in the schedule.

By reconstructing the partially ordered set P in figure 3.1 into a bipartite graph $G = [S, T; A]$ consisting of 2n nodes $S = \{t_1, t_2, \ldots, t_{10}\}$ and $T = \{t_1, t_2, \ldots, t_{10}\}$ and defining arcs by the rule: $(t_i, t_j) \in A$ if and only if $t_i \rangle t_j$ we can find the minimal chain decomposition (cover) by using an algorithm which determines the maximal number of independent admissible cells.

3.12 - MINIMAL DECOMPOSITION ALGORITHM

To find the minimal decomposition in the partially ordered set we can use the labelling process for maximal flows. First the bipartite graph G is depicted in an array format. The columns and rows of the array are the elements of S and T respectively. Inadmissible cells, cells where no arc (t_i, t_j) exists, are marked with an X; while admissible cells, cells where arcs (t_i, t_j) do exist, are left blank. From this format the algorithm can be quickly executed.

The goal of the algorithm is to place as many 1's as possible in admissible cells. The only restriction is that no more than one 1 can be in any row or column. The algorithm proceeds as follows:

Step 1 - Start with any feasible placement of 1's in the array. (recall only one 1 can be in any row or column).

Step 2 - Label all rows which do not contain a 1.

Step 3 - Select an unscanned, labelled row and scan it for admissible cells.

Step 4 - Label the columns which contain the admissible cells found in step 3 with the row number being scanned. If a column is already labelled do not relabel it.

Step 5 - repeat steps 3 and 4 until all labelled rows have been scanned and the appropriate columns have been labelled.

Step 6 - Select an unscanned, labelled column and scan it for a 1. If a 1 is found label the row containing the 1 with the column number.

Step 7 - Repeat step 6 until all labelled columns have been scanned and the appropriate rows have been labelled.

Step 8 - Repeat steps 3 through 7 until one of the following occurs:

Case 1 -- A labelled column is scanned and no 1 is found. This is called a *breakthrough*.

Case 2 -- No more row or column labels are possible (*no breakthrough*).

STOP.

If Case 1 occurs then the current placement of 1's is not maximal and the chain decomposition is not minimal. The number of 1's in the array can be increased by one in the following manner:

Step 1 - In the just labelled column which has no 1 place a 1 in the row indicated by the column's label.

This row will now have two 1's in it since the row was previously scanned. The other 1 must now be moved somewhere else in its column since no more than a single 1 can be in any row. The conflicting 1 is moved in the following manner:

Step 2 - Move the conflicting 1 to the row indicated by its column label. Repeat this step for all subsequent conflicts until a conflicting 1 is moved to a row in which no conflict occurs.

STOP.

This method will always increase the number of 1's in the array by one because eventually a conflicting 1 will be moved into one of the original scanned rows which did not have a 1.

After increasing the number of 1's in the array all labels are removed and the entire process is repeated until Case 2 occurs. When Case 2 occurs no more row or column labels are possible and there is no breakthrough. At this point the number of 1's is maximal and the minimal chain decomposition is reached.

At this point an example of the algorithm is appropriate. Using the partially ordered set in figure 3.1 the array representation can be seen in table 3.2. Like the bipartite graph the array depiction includes all feasible arcs (t_i, t_j) $t_i \varepsilon S$, $t_j \varepsilon T$. The feasible arcs are the admissible cells(those without an X).

Table 3.2 - Array Depiction of the Bi-Partite Graph

Tasks	1	2	3	4	5	6	7	8	9	10	
1	X	X	X	X	X						
2	X	X	X	X							
3	X	X	X								
4	X	X	X	X	X	X					
5	X	X	X	X	X	X	X		X		
6	X	X	X	X	X	X			X		
7	X	X	X	X	X	X	X	X	X	X	
8	X	X	X	X	X	X	X	X	X		
9	X	X	X	X	X	X	X	X	X		
10	X	X	X	X	X	X	X	X	X	X	
											Labels

The next array in table 3.3 shows the results of the labelling algorithm. Initially six 1's are placed in the admissible cells(Step 1). No other 1's can be added so far since any addition would violate the restriction of one entry per line. The associated chain decomposition is read from the array to be {1, 6, 10; 2, 5, 8; 3, 4, 7; 9}. We will now determine if this decompostion into 4 chains is minimal. (Step 2) Rows 7, 8, 9, and 10 do not have 1's and are labelled (indicated by ----). (Step 3) Row 8 is scanned yielding one admissible cell. (Steps 4) The column containing this cell is labelled with the number 8. (Step 5) Scanning row 9 yields no new labels. (Step 6,7) Column 10 is scanned for a 1 which is found in row 6. Row 6 is labelled by the column number 10. (Step 8) Repeating steps 3 through 7 for row 6 labels columns 7 and 8. These columns in turn label rows 4 and 5 respectively. Scanning row 4 labels column 9(note:

column 10 is already labelled), and scanning row 5 results in no new labels. Finally, scanning column 9 yields no 1. Thus we have a breakthrough, indicated by a O, and the process is complete.

Table 3.3 - Labelling Scheme

Tasks	1	2	3	4	5	6	7	8	9	10	
1	X	X	X	X	X	1					
2	X	X	X	X	1						
3	X	X	X	1							
4	X	X	X	X	X	X	1		O		7
5	X	X	X	X	X	X	X	1	X		8
6	X	X	X	X	X	X			X	1	10
7	X	X	X	X	X	X	X	X	X	X	----
8	X	X	X	X	X	X	X	X	X		----
9	X	X	X	X	X	X	X	X	X		----
10	X	X	X	X	X	X	X	X	X	X	----
							6	6	4	8	Labels

Using the two step procedure to resolve conflicts a 1 is placed in the breakthrough cell - row 4, column 9. The conflicting 1 in row 4 is in column 7. This 1 is moved to the row indicated by column 7's label - row 6. Similarly the conflicting 1 in row 6 is moved to row 8. Since row 8 is the original labelled row no conflict exists. Table 3.4 displays the conflict resolution and the new configuration of 1's. Notice the number of 1's has increased by one from six to seven.

Table 3.4 - Conflict Resolution

Tasks	1	2	3	4	5	6	7	8	9	10	
1	X	X	X	X	X	1					
2	X	X	X	X	1						
3	X	X	X	1							
4	X	X	X	X	X	X			1		
5	X	X	X	X	X	X	X	1	X		
6	X	X	X	X	X	X	1		X		
7	X	X	X	X	X	X	X	X	X	X	----
8	X	X	X	X	X	X	X	X	X	1	10
9	X	X	X	X	X	X	X	X	X		----
10	X	X	X	X	X	X	X	X	X	X	----
										9	Labels

Repeating the entire process on the new configuration yields rows 7, 9, and 10 as not having any 1's. The only admissible cell is in row 9. However, the labelling process quickly results in no breakthrough. Thus the configuration in table 3.4 is maximal. The new chain decomposition from the array is {1, 6, 7: 2, 5, 8, 10: 3, 4, 9}. This decomposition into three chains is minimal by Dilworth's theorem. The bold lines in figure 3.2 are the minimal chains in the partially ordered set resulting from the algorithm. According to Papdimitriou and Steiglitz, labelling algorithms of this type for the bipartite matching problem require $O(\min (|S|,|T|) \cdot |A|)$ time[50].

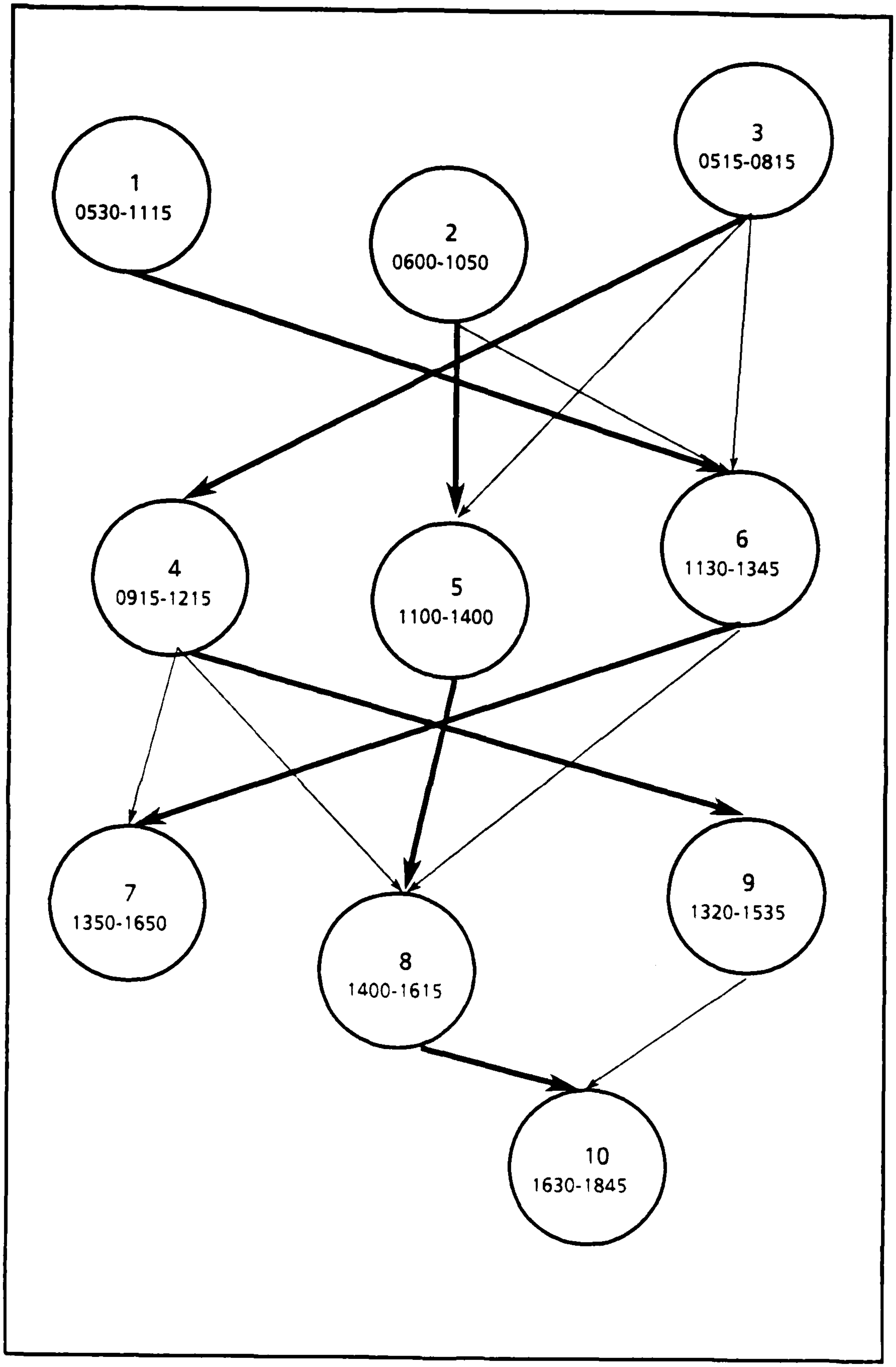


Figure 3.2 - Network depiction of the Dilworth chains

3.13 - EXPLOITING SPECIAL STRUCTURE

On occaision the Level 1 scheduling problem exhibits special structure which can be exploited in determining the minimal chain decomposition. If the members of the partially ordered set can be renumbered so that $t_i \leq t_j$ implies that the predecessors of t_i are included in the predecessors of t_j then the *Staircase rule* applies. Unlike the labelling algorithm just described, this simple rule requires no iterations. If the array can be rearranged such that the set of all inadmissible cells has a staircase form, the rule applies.

> **Staircase Rule**: *Select any admissible cell that borders the staircase of inadmissible cells and place a 1 in it. Delete the row and column and repeat the process.*[FF 65]

By exchanging the rows and columns in the example array, the array displays the staircase form (table 3.5).

Table 3.5 - Staircase Form

Tasks	1	2	3	4	5	6	9	7	8	10	
3	X	X	X	1							
2	X	X	X	X	1						
1	X	X	X	X	X	1					
4	X	X	X	X	X	X	1				
6	X	X	X	X	X	X	X	1			
5	X	X	X	X	X	X	X	X	1		
8	X	X	X	X	X	X	X	X	X		
9	X	X	X	X	X	X	X	X	X	1	
10	X	X	X	X	X	X	X	X	X	X	
7	X	X	X	X	X	X	X	X	X	X	

Using the staircase rule, starting at the top left, 1's are placed at each border point. The corresponding minimal chain decomposition can be seen directly to be {1, 6, 7; 2, 5, 8: 3, 4, 9, 10}. These chains are not the same chains determined by the labelling process. I will discuss how to decide between alternative solutions in the next section. Obviously, this structure should be exploited whenever possible since it quickly determines the minimal decomposition in O(N) steps. As was mentioned earlier, the staircase form may exist in the Level 1 problem.

3.2 - MAXIMUM FLOW FORMULATION

An alternative formulation for the Level 1 problem is an adaptation of the Dilworth Chain decomposition. The formulation, which uses a maximum flow procedure, is the basis for the vehicle scheduling algorithm developed by Bodin, Rosenfield, and Kydes in UCOST[14]. Using the bipartite graph representation G = [S, T; *A*] discussed earlier, figure 3.3 shows the sample problem constructed as a bipartite graph. The node sets S and T each contain all the nodes of the partially ordered set, P. Notice in this configuration that all feasible arcs (t_i, t_j) $t_i \varepsilon S$, $t_j \varepsilon T$ are included in the graph unlike the network depiction of the partially ordered set in figure 3.1. In addition to S and T two nodes are added to the network - a super source, s, and a super sink, t. Each node in S is connected to the super source, and each node in T is connected to the super sink. The capacity on all arcs connected to the source and sink is 1. Finally, one additional directed arc from the sink to the source is added to complete the circulation.

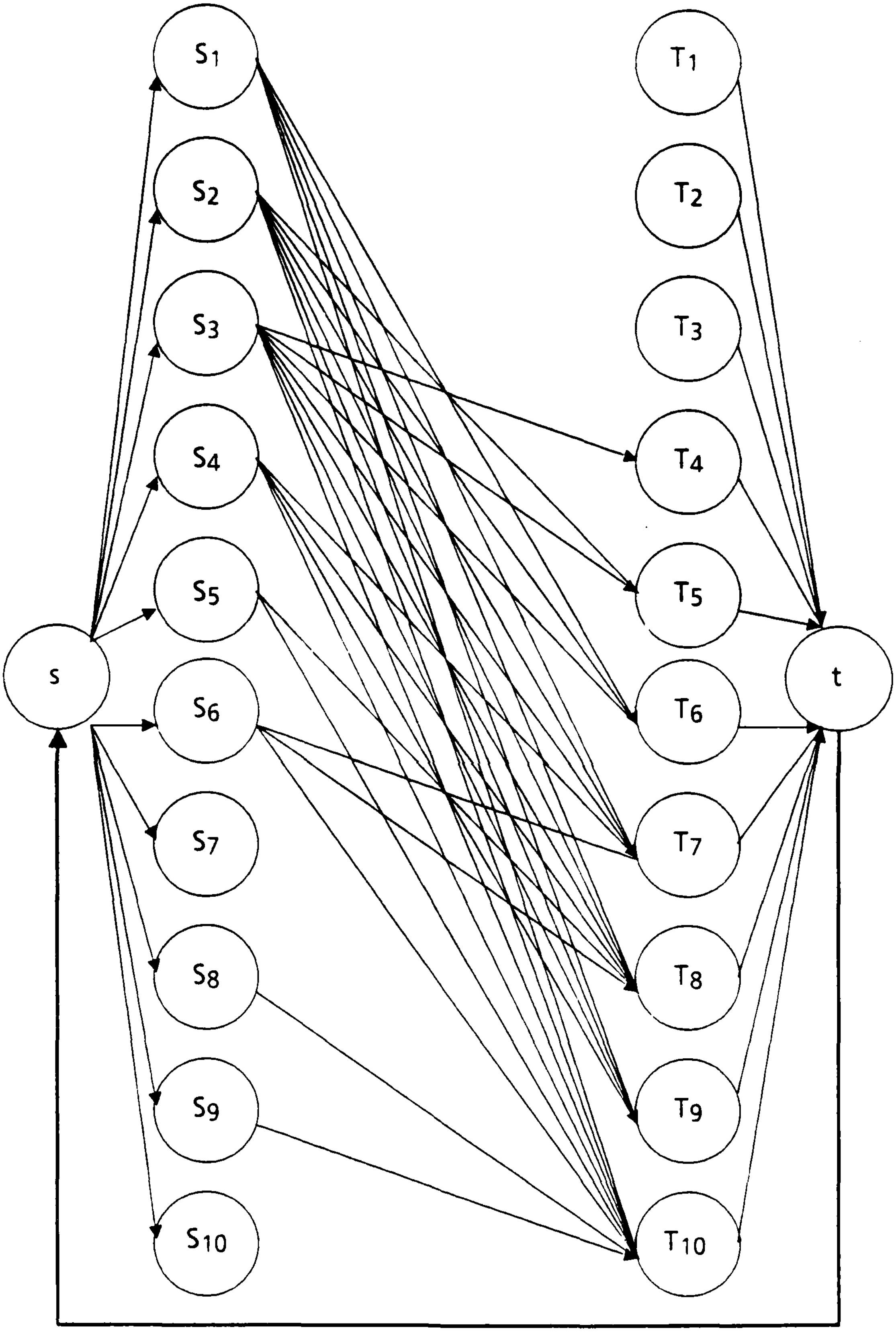


Figure 3.3 - Maximum Flow Representation

Using x_{ij} to represent the flow on arc (t_i, t_j) the formulation becomes

(3.4) Maximize x_{ts}

(3.5) s.t. $\Sigma_j x_{ij} - \Sigma_k x_{ki} = 0 \qquad i,j,k \varepsilon S,T,s,t$

(3.6) $0 \leq x_{ij} \leq 1$, integer

Equation 3.5 is the flow balance equation requiring that the flow out of a node minus the flow into the node must equal zero, and 3.6 captures nonnegativity and flow capacity restriction for each arc. Finally, we let

$$D = \{ x_{ij} \varepsilon A;\ i \varepsilon S, j \varepsilon T;\ x_{ij} = 1 \}.$$

The arcs in D are used to determine the minimum number of disjoint chains in G. By the Konig - Egevary Theorem

(3.7) minimum number of chains $= |S| - |D|$.

Using the optimal solution to the max flow formulation the arcs in D can be used to determine the task chains for each instructor. Let (S_i, T_j) be an arc in D where S_i is chosen so there is no flow into T_i(i.e. S_i is the start of a schedule). Eliminate arc (S_i, T_j) from D and find the arc $(S_j, T_K) \varepsilon$ D. This arc is linked to the previous one since nodes S_j and T_j represent the same task. Continue the procedure until an arc is found (S_m, T_n) such that there is no flow out of node S_n. The chain sequence

$$(S_i, T_j), (S_j, T_k), \ldots, (S_m, T_n)$$

represents a feasible schedule associated with tasks t_i, t_j, t_k, . . ., t_n in the partially ordered set P. The process is repeated in this manner until all |S| - |D| chains are found by eliminating all arcs in D. The max flow solution to the sample problem is in figure 3.4. The arcs in D are

$$D = (S_1,T_6), (S_2,T_5), (S_3,T_4), (S_4,T_9), (S_5,T_8), (S_6,T_7), (S_8,T_{10}).$$

The number of chains is |S| - |D| = 10 - 7 = 3. The sequence in each chain is {1,6,7}, {2,5,8,10}, and {3,4 9}.

A theorem of Hopcroft and Karp[38] shows the bipartite matching problem can be solved in $O(|V|^{0.5} \cdot |A|)$ time. It is the asymptotically fastest algorithm known for the bipartite matching problem. The fact that this algorithm is a special case of the max-flow algorithm applied to a simple network was first pointed out by Even and Tarjan[22].

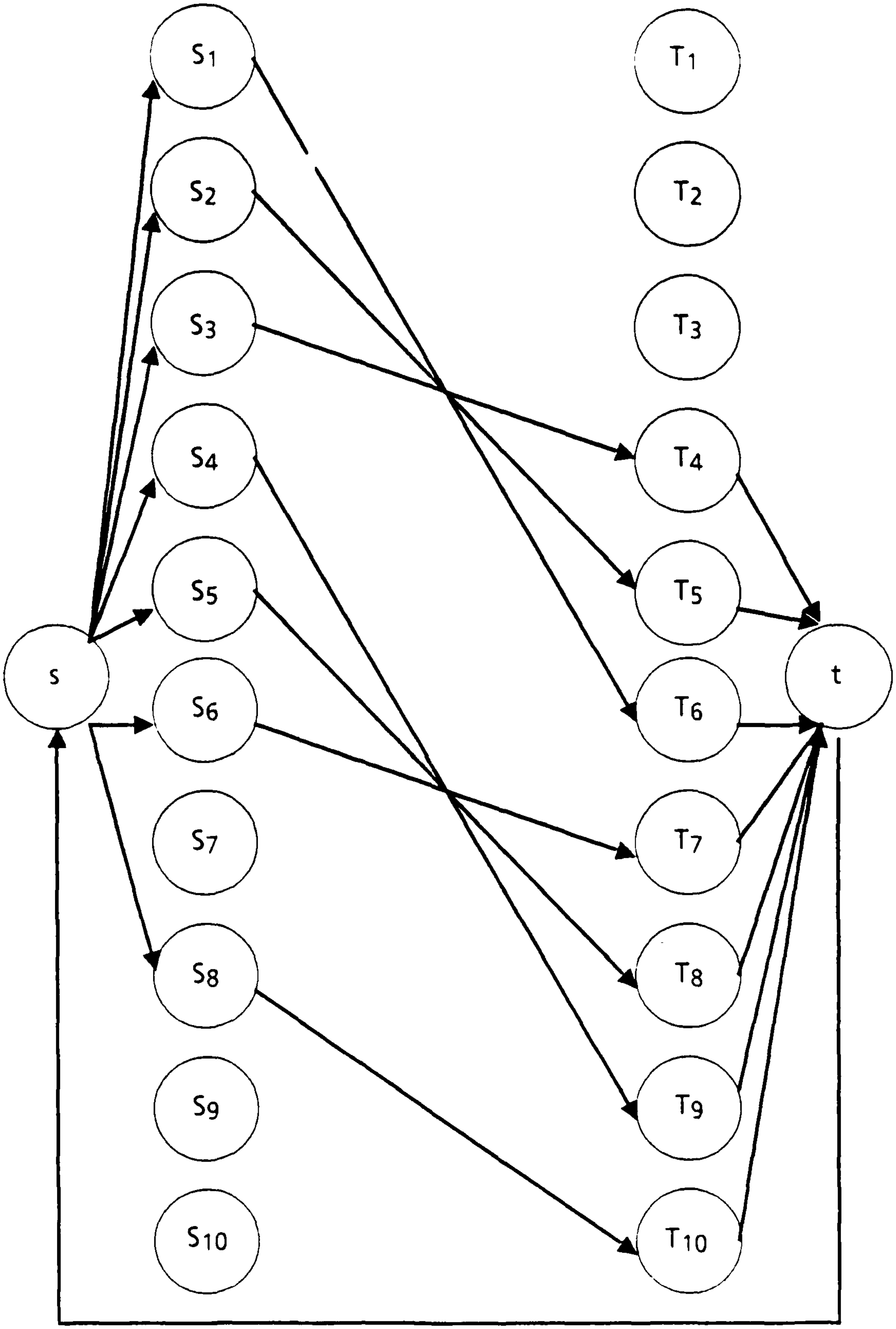


Figure 3.4 - Maximum Flow Solution

3.3 - MINIMUM COST FLOW FORMULATION

After using Dilworth's decomposition to determine the minimum number of instructors to accomplish all the tasks, the next step is to use these instructors in the most efficient or least cost manner. As was shown in the sample problem the minimal chain solution might not be unique. The Dilworth decomposition only provides the minimal number of chains and does not attempt to find a "best" solution in terms of some cost function. To find the least cost, minimal chain decomposition I will use the minimum cost network flow formulation. The use of this formulation to solve vehicle scheduling problems was first developed by Dantzig and Fulkerson[19].

Bradley, Hax, and Magnanti[15]; Papadimitriuo and Steiglitz[50] and others discuss this formulation in detail. To transform the Level 1 problem into a *minimum cost flow* problem two nodes must be added to the partially ordered set P. The first node , s, is the source node which can be interpreted as the pool of instructors. The second node, b, is the sink node which can be interpreted as the officer's club bar. An arc is added from s to each task node $t_1, t_2, \ldots, t_n$ in P and from each task node to the sink node t. In all (2·num. nodes) arcs are added. The cost assigned to each of these arcs is 0. The cost, c_{ij}, for each arc in P is proportional to the idle time between tasks t_i, t_j where

$$(3.8) \qquad \text{Idle Time} = S_j - B_j - (E_i + DB_i).$$

For the sample problem I will assign a cost of 1 for every 15 minute block of idle time(or portion thereof) between two tasks. To prevent the number of arcs from getting too large and to align the problem with realistic scheduling, arcs representing idle time greater than a user specified amount (α) will be excluded from the network. This will eliminate solutions with excessive idle time for an instructor. For the sample problem, I will restrict the idle time to 3 hours or less. Arcs longer than 2 hours will not be included in the network. The associated cost(idle time) matrix for the feasible arcs is in table 3.6. The resulting minimum cost flow network is in figure 3.4. The source node has a supply of 3 instructors, and the sink node has a demand of 3 instructors. The objective is to flow the 3 instructors through all the task nodes to the sink node with the least total idle time between tasks. The following linear programming formulation solves the problem.

(3.9) Minimize $\Sigma_i \Sigma_j \ c_{ij} x_{ij}$

(3.10) s.t. $\Sigma_j x_{ij} - \Sigma_k x_{ki} = b_i \quad i \varepsilon P, s, t$

(3.11) $\Sigma_j x_{ij} = 1 \quad j \varepsilon P$

(3.12) $\Sigma_i x_{ij} = 1 \quad i \varepsilon P$

(3.13) $0 \leq x_{ij} \leq 1$, integer

$$x_{ij} = \begin{matrix} 1 & \text{if arc (i,j) is in the solution} \\ 0 & \text{otherwise} \end{matrix}$$

Equation 3.10 is the flow balance equation requiring that the flow out of a node minus the flow into the node must equal the net supply(demand) at the node. b_i equals zero for all nodes in P. For s,t b_i equals + or - the minimal number of instructors from the Dilworth decomposition. Equations 3.11 and 3.12 ensure that a task node is covered only once in the solution.

Table 3.6 - Cost Matrix for Feasible Arcs

Tasks	1	2	3	4	5	6	7	8	9	10
1	X	X	X	X	X	1			8	
2	X	X	X	X	1	3				
3	X	X	X	4						
4	X	X	X	X	X	X	7	7	5	
5	X	X	X	X	X	X	X	0	X	
6	X	X	X	X	X	X	1	1	X	
7	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	1
9	X	X	X	X	X	X	X	X	X	4
10	X	X	X	X	X	X	X	X	X	X

The solution to the minimization problem is the chain decomposition {1, 6, 7; 2, 5, 8, 10; 3, 4, 9} which is the same as the decomposition determined by the max flow labelling algorithm.

The resulting costs are

Chain	Cost
1-6-7	2
2-5-8-10	2
3-4-9	9
Total	13

The alternative decomposition {1, 6, 7; 2, 5, 8; 3, 4, 9, 10} determined by the staircase rule results in a total cost of 16. Thus, by using this two step process the solution provides the most efficient way, in terms of idle time, to schedule the minimum number of instructors needed to accomplish all the tasks.

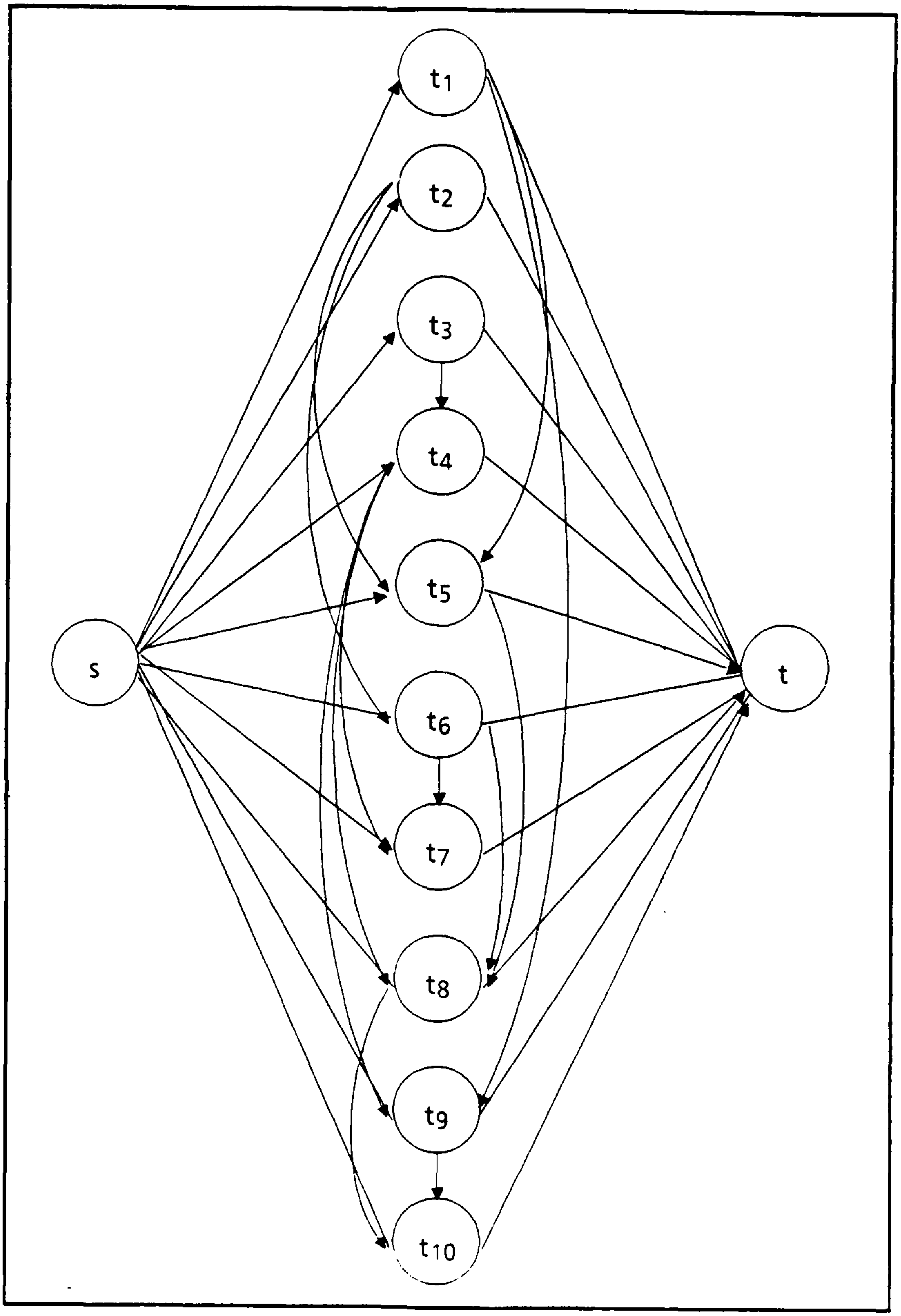


Figure 3.5 - Minimum Cost Network Flow Depiction

3.4 - HEURISTIC MODIFICATIONS

After determining the optimal solution to the relaxed Level 1 problem, the difficulty of meeting the relaxed constraints can now be addressed. The relaxed solution technique cannot directly accomodate the 12 hour duty day or the maximum of 3 tasks per instructor limitations[4]. This is evident in the solution to the relaxed sample problem in table 3.7.

Table 3.7 - Instructor Duty Schedule

Task Chain	Task Descriptions	No. Tasks	Duty Day
1 - 6 - 7	SOF - simulator - aircraft	3	0530 - 1650
2 - 5 - 8 - 10	RSU - Aircraft - simulator-sim.	**4**	**0600 - 1845**
3 - 4 - 9	Aircraft - aircraft - simulator	3	0515 - 1535

The solution violates both constraints(indicated by bold type). Of the two constraints the duty day restriction is the least binding and might possibly never be a factor. This occurs because the minimum cost flow formulation will attempt to pack the tasks in each chain with the minimum amount of idle time between tasks. Since the vast majority of tasks require 3 hours or less total time to complete, and since 3 tasks is maximum, the average time performing tasks will be less than 9 hours per instructor. This leaves a full 3 hours in the duty day to absorb the two idle periods between tasks. Thus, it is highly unlikely that the duty day constraint will ever be violated by a chain containing 3 or less tasks.

Although table 3.7 indicates the relaxed solution violates both constraints, reducing the number of tasks in all chains to 3 or less will alleviate both problems. Obviously, the scheduler must add one more instructor to the schedule. The process of trimming chains and adding instructors is an interactive one. The squadron scheduler may have several options available. For instance, in the sample problem the scheduler could trim the first or last task of the long chain, or he could divide the chain into two chains of 2 tasks each. All of these divisions will result in slightly different total idle time. The scheduler considers various items such as workload balancing, seniority, time-off requests, etc. to determine the best schedule. Also, the scheduler could add lower priority tasks such as medical appointments, training sessions, etc. which are time flexible and can fit anywhere in the schedule.

Finally, if several chains of length 4 or more exist in the schedule the scheduler can incrementally increase the minimum number of instructors(if available) for the minimum cost flow formulation. The solution will always use the added instructors since the arc costs from P to s and t is zero. The least cost solution for the sample problem using 4 instructors is in table 3.8.

Table 3.8 - Least cost schedule

Task Chain	Task Descriptions	No. Tasks	Chain Cost	Duty Day
1 - 6 - 7	SOF - simulator - aircraft	3	2	0500 - 1650
2 - 5 - 8	RSU - Aircraft - simulator	3	1	0530 - 1615
3 - 4	Aircraft - aircraft	2	4	0445 - 1215
9 - 10	Simulator - simulator	2	4	1250 - 1845

Notice that the formulation did not merely trim a task off the longest chain to obtain the optimal solution. The highest cost arc is between tasks 4 and 9. Since there is a feasible arc from task 9 to task 10 with a cost less than 5, the formulation trims off task 10 from the long chain and eliminates the highest cost arc(4,9) by linking tasks 9 and 10. The resulting total cost is 11. This cost is less than the relaxed solution since the fourth instructor is added.

There are several other feasible solutions to the problem. Some of the alternate solutions have only slightly higher total idle time. A list of the alternate feasible solutions is in table 3.9.

Table 3.9 - Alternative solutions

Feasible Decompositions		Total Cost
Number	Chains	
1	{1,6,7},{2,5,8},{3,4},{9,10}	11
2	{1,6,7},{2},{5,8,10},{3,4,9}	12
3	{1,6,7},{2,5,8},{10},{3,4,9}	12
4	{1,6,7},{2,5,8},{3},{4,9,10}	12
5	{1,6,7},{2,5},{8,10},{3,4,9}	13

Notice that the solution which merely subdivides the infeasible chain into two feasible chains(decomposition no. 5) results in the highest cost. This shows that the "obvious" solution might not always be the best solution. However, this is not to imply that the scheduler will always use the least cost solution. Based on the scheduler's input some other slightly different yet higher cost solution may better fit the needs of the squadron.

After determining the task schedule for the entire squadron the scheduler assigns each task chain to individual instructors based on unit needs, instructor requests, and chain make-up. The scheduler then makes a final adjustment to the duty day. The lenght of day determined by the algorithm only includes the time required to accomplish all tasks. It does not include the individual instructor's initial briefing upon arrival at the squadron. The initial briefing occurs 30 minutes prior to the first activity of the day for each instructor. Thus, after determining an instructor's schedule, the squadron scheduler adds 30 minutes to the beginning of each chain to obtain the instructor report times. This process is equivalent to the vehicle scheduling procedure of adding vehicle deadheading time from a garage. The duty day time in table 3.8 includes the briefing time for each instructor.

CHAPTER 4

STUDENT SCHEDULING PROBLEM (LEVEL 2)

4.0 - INTRODUCTION

After the squadron scheduler finds a feasible set of task assignments for the instructors in the Level 1 problem, the unit schedulers can address the student scheduling problem. Although similar network structures exist in both problems, the Level 2 problem has a vastly different population of pilots, set of constraints, and an altogether different set of objectives.

As was mentioned in chapter 1 the student scheduling problem is addressed on a daily basis by the unit schedulers. The unit scheduler is responsible for equitably scheduling the unit's students to maintain proper timeline flow and to meet Air Training Command regulations and squadron guidelines[57]. Having received the Level 1 scheduling output which describes the daily activities of the unit's instructor force, each of the six unit schedulers attempts to "optimally" match students and instructors to produce an effective, smooth flowing schedule for the unit. Each unit schedules its students independently as the students progress through the various stages of training.

4.01 - CONSTRAINTS

Recall that the constraints on the Level 2 problem go beyond those of the Level 1 problem. The student population has the same duty day and maximum number of tasks restrictions as the instructor force, and additional continuity constraints which limit the task types for each student. Thus, unlike the Level 1 problem which is concerned only with the precedence relationships among tasks, the Level 2 problem involves the proper matching of students and instructors. This matching concerns whether a given student is allowed to perform a task with the instructor previously assigned to that task by the Level 1 scheduler. In the vehicle scheduling or production planning context, this restriction is equivalent to determinimg whether a particular vehicle/machine has the proper capacity, tooling, etc. to perform an available task. In short, the student pilots do not form a homogeneous group of individuals. A summary of the Level 2 problem constraints are:

1) 3 tasks maximum per student per day
2) 2.75 hours minimum between task starts(turn time)
3) 12 hour maximum duty day
4) Limited instructor matchings available

4.02 - SAMPLE PROBLEM

To aid in visualizing the formulation of the Level 2 problem I will continue to use the sample problem of the previous chapter. Assume that the ten tasks are all assigned to four instructors in a single unit. Assume also that the unit scheduler has six students available for assignment to the tasks. Actual unit scheduling problems usually involve fifteen instructors, thirty students, and over forty tasks per day(see appendix B). Table 4.1 is a summary of the results of the instructor scheduling problem. It contains the task list with the instructor assignments determined by the Dilworth/Min-cost flow procedure. Column 4, the adjacency list, is an alternative representation of the array in table 3.2 listing all feasible matchings of tasks in the schedule.

Table 4.1 - Instructor -Task Assignment Summary

Task	Description	Instructor	Task Adjacency List
1	RSU	4	6 - 7 - 8 - 9
2	SOF	3	5 - 6 - 7 - 8 - 9
3	Aircraft	1	4 - 5 - 6 - 8 - 9
4	Aircraft	1	3 - 7 - 8 - 9 - 10
5	Aircraft	2	2 - 3- 8 - 10
6	Simulator	4	1 - 2 - 3 - 7 - 8 - 10
7	Aircraft	4	1 - 2 - 4 - 6
8	Aircraft	2	1 - 2 - 3 - 4 - 5 - 6 - 10
9	Simulator	1	1 - 2 - 3 - 4 - 10
10	Simulator	2	4 - 5 - 6 - 8 - 9

Table 4.2 shows the preference list for the student - instructor matchings. In it the six students are listed with their instructors in order of preference, the most preferred being the student's assigned instructor.

Table 4.2 - Student Preference List

Student	Instructor preference most--------least		
1	4	3	1
2	2	3	
3	1		
4	2	4	
5	4		
6	1	2	

4.1 - PROBLEM FORMULATION

The goals of the unit scheduler are quite different than those of the squadron scheduler. The squadron scheduler's goal is to maximize the use of each instructor during the day thus providing the best work environment for the instructor. The unit scheduler, on the other hand, wishes to disperse daily activities among students so that no student is accomplishing several tasks in a given schedule. This allows a better training environment for the students as adequate preparation time is alloted for each activity. As a result, the objective of the unit scheduler is threefold. First, the unit scheduler attempts to schedule each student at least once a day if possible. This allows a good flow of activity for each student and prevents students from being idle for an entire day.

The second objective is to provide proper continuity of instruction for each student. If a student is involved with too many instructors he/she may receive inconsistent training due to the variety of techniques used by each instructor. Each student is assigned to one instructor for training and administrative purposes. In addition to the assigned instructor, each student receives training from one or more additional instructors as he/she progresses through the program. For continuity purposes it is best if a student's assigned instructor conducts roughly sixty percent of the training. The remaining forty percent is conducted by the alternate instructors.

The third objective of the unit scheduler clearly shows the difference between Level 2 and Level 1 scheduling. Whereas the squadron scheduler's goal is to minimize idle time for the instructors, the unit scheduler attempts to maximize idle time for the students. Due to the demanding nature of the training it is very difficult for a student to maintain a competent level of performance if there is little or no idle time between tasks. Both the safety of the mission and the student's ability to receive training are hindered if no rest periods exist between activities.

4.11 - SIMPLE ASSIGNMENT PROBLEM

The nature of the unit scheduler's objectives indicate that the Level 2 problem falls into the general category of assignment problems. Most textbooks on Mathematical Programming, Networks, and Combinatorial Optimization cover solution techniques to simple assignment problems[15,50]. The assignment problem is a more involved version of the bipartite matching problem discussed in chapter 3. In addition to the bipartite graph, $G=\{S,T;A\}$, a number $w_{ij} \geq 0$ for each arc $(S_i,T_j) \varepsilon A$

represents the weight or disutility of the matching. We then find the matching with the minimum amount of weight. Figure 4.1 shows the sample problem as an assignment problem. The bipartite graph is comprised of the set of students, S, and the set of instructor-task matchings, T, from table 4.1. The asssigned instructor for each task is adjacent to the task in figure 4.1. The set of connecting arcs in *A* are those which link each student in S to a feasible instructor/task in T. The weights on the arcs capture the instructor preference objective of the unit scheduler. Less weight is given to more preferred student-instructor pairings. Arcs linking a student with his/her assigned instructor have the minimum weight, while alternative pairings are given more weight according to the preference order for each student. For example, student one in the sample problem has an instructor preference list of 4,3,1 in table 4.2. Thus any arc in *A* connecting student one(node 1 in S) with a task performed by instructor four(nodes 1,6,7 in T) will have a weight of 1 since the instructor is the student's most preferred instructor. Similarly, arcs in *A* connecting student one with tasks in T performed by instr ıctors three or one will have proportionately greater weights respectively. This weighting is accomplished for all arcs in *A*. Figure 4.1 shows the feasible arcs for the sample problem. There are 29 feasible arcs in all.

Finally, to convert the Level 2 problem into the assignment problem two adjustments must be made to the graph, G. First, the number of student nodes in S must equal the number of task nodes in T. Since this is usually not the case in most assignment problems, artificial or 'dummy' nodes must be added. In the sample problem four 'dummy' student nodes(labelled $D_1,...,D_4$ in 4.1) are added. The second adjustment requires the graph G be 'complete'. That is, all nodes in S must be connected to all nodes in T. Since the arcs in *A* do not accomplish this, artificial arcs are added to G to make the graph complete.

These additional arcs are represented by dashed lines in figure 4.1. To prevent figure 4.1 from becoming too cluttered only the artificial arcs from student node 3 and 'dummy' node D_2 are included in the depiction. Obviously, these arcs create infeasible student-instructor matchings because all feasible matchings are contained in the set A. To prevent these infeasible matchings from entering the solution a very large weight, M, is assigned to each artificial arc. Thus, the minimization algorithm will attempt to eliminate the high weight arcs from the solution. If a prohibited student-instructor matching cannot be eliminated from the optimal solution then the problem, as stated, is infeasible.

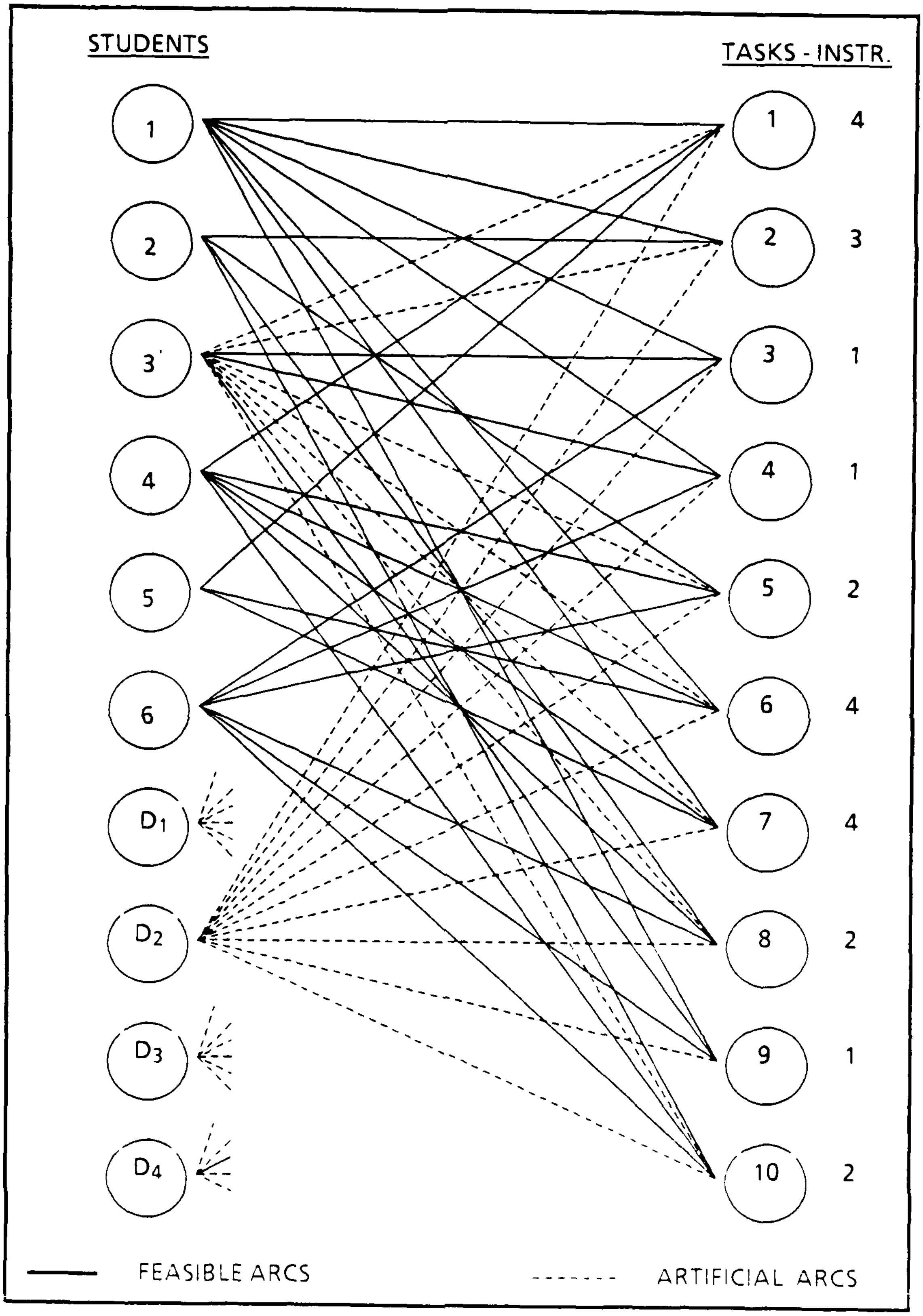


Figure 4.1 - Sample Assignment Problem

If we let the decision variable

$$x_{ij} = \begin{matrix} 1 \text{ if student i is assigned to instructor/task j} \\ 0 \text{ otherwise} \end{matrix}$$

the optimal assignment results from the following formulation:

$$\text{(4.1)} \qquad \text{Minimize} \quad \Sigma_i \Sigma_j \; w_{ij} x_{ij}$$

$$\text{(4.2)} \qquad \text{Subject to} \quad \Sigma_j x_{ij} = 1 \qquad (i = 1, 2, \ldots, n)$$

$$\text{(4.3)} \qquad \Sigma_i x_{ij} = 1 \qquad (j = 1, 2, \ldots, n)$$

$$\text{(4.4)} \qquad 0 \le x_{ij} \le 1, \text{integer} \qquad (i,j = 1, 2, \ldots, n)$$

Equation 4.2 allows each student to be assigned to exactly one instructor/task. Equation 4.3 indicates that each task is to be performed by exactly one student.

As written, the assignment problem is formally an integer program since the decision variables x_{ij} are restricted to be zero or one. However, by replacing the constraint in 4.4 with $x_{ij} \geq 0$ and having a supply of one unit at each node in S and a demand of one unit at each node in T, the problem becomes a transportation problem. Due to the theory of total unimodularity[50] this formulation will always have integer solutions and a network flow algorithm will solve the integer problem directly.

The optimal simple assignment solution to the sample problem is indicated by the bold lines in figure 4.2. Notice that each of the six students is assigned to a task accomplished by his her assigned instructor.

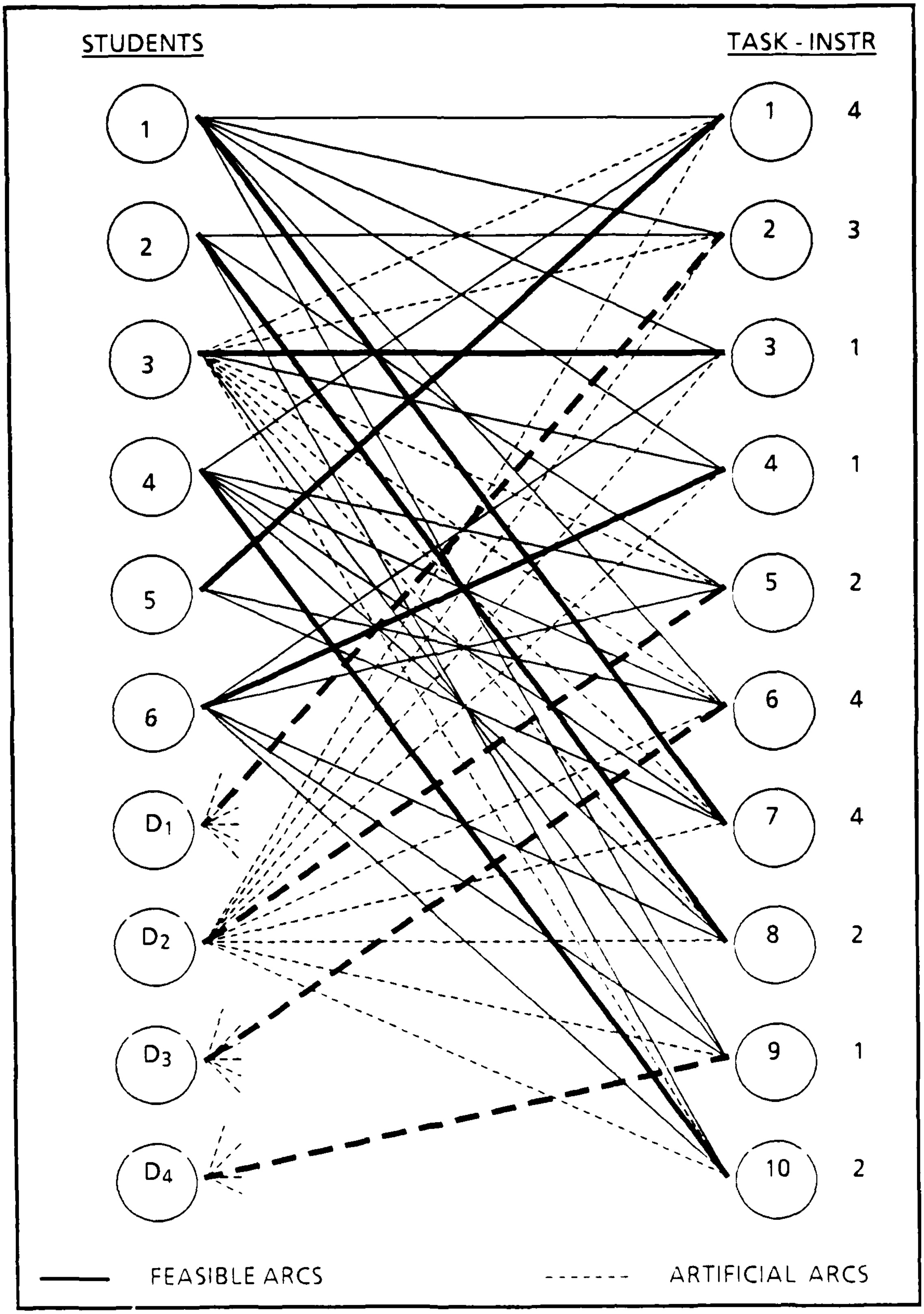


Figure 4.2 - Optimal Solution to Sample Assignment Problem

Although the simple assignment formulation quickly accomodates the unit scheduler's first two objectives by (1) attemping to assign each student a task, and by (2) scheduling events to include the arcs in G with the lowest total weights(most preferred pairings), it cannot accomodate the sequential nature of the tasks. Recall that the tasks in T have an ordered relationship which indicates that more than one task can be accomplished by a particular student in S. However, the simple assignment formulation allows only a one-to-one matching in the optimal solution due to equations 4.2 and 4.3. Thus the formulation assigns the most preferred set of tasks to the students and assigns the remaining tasks to the dummy nodes. There is no capability of assigning a remaining task to a student even though it may be feasible(i.e. an arc exists from t_i to t_j in the partially ordered set). For example, student 6 can accomplish both task 4 and task 9 (which was assigned to a dummy node by the simple assignment problem) with his/her assigned instructor. In the following three sections I will discuss formulations which can accomodate the ordered relationship.

4.12 - INTEGER PROGRAMMING

One approach to solving the Level 2 problem optimally is through the use of integer programming. The student scheduling problem is similar in form to the capital budgeting problem discussed in literature[15]. The decision variables in the capital budgeting problem x_{ij} are taken to be 0 or 1 and represent a go / no-go decision with $x_{ij} = 1$ indicating that student i is matched with task j. Assuming w_{ij} is the weight or cost of the matching the problem can be stated as:

$$\text{(4.5)} \qquad \text{Minimize} \quad \Sigma_i \Sigma_j w_{ij} x_{ij} \qquad (i=1,\dots, m), (j=1,\dots, n)$$

$$\text{(4.6)} \qquad \text{Subject to:} \quad \Sigma_j x_{ij} \leq b_i \qquad (i=1, 2, \dots, m)$$

$$\text{(4.7)} \qquad \Sigma_i x_{ij} = 1 \qquad (j=1, 2, \dots, n)$$

$$\text{(4.8)} \qquad x_{ij} = 0 \text{ or } 1$$

Equation 4.6 ensures that student i does not perform more than a pre-specified total number of tasks, b_i (i.e. 1, 2, or 3 maximum). Equation 4.7 requires that all tasks be covered exactly once.

In this simple form the integer programming formulation fails to capture the conflicting nature of the tasks. That is, it is possible to assign two simultaneous tasks to a single student. It is necessary, then, to add multiple-choice constraints to the formulation based on the information in the adjacency list. For example, in the sample problem only one of tasks 1,2,3 may be accomplished by a single student. The constraint

$$\text{(4.9)} \qquad x_{i1} + x_{i2} + x_{i3} \leq 1 \qquad (i=1,2,\dots,m)$$

must be added to capture this logical constraint. Other logical constraints for the sample problem are:

(4.10) $x_{i1} + x_{i4} + x_{i5} \le 1 \qquad (i=1,2,\ldots,m)$

(4.11) $x_{i2} + x_{i4} \le 1 \qquad (i=1,2,\ldots,m)$

(4.12) $x_{i6} + x_{i5} + x_{i6} \le 1 \qquad (i=1,2,\ldots,m)$

(4.13) $x_{i6} + x_{i9} \le 1 \qquad (i=1,2,\ldots,m)$

(4.14) $x_{i5} + x_{i7} + x_{i9} \le 1 \qquad (i=1,2,\ldots,m)$

(4.15) $x_{i8} + x_{i7} + xi9 \le 1 \qquad (i=1,2,\ldots,m)$

(4.16) $x_{i7} + x_{i10} \le 1 \qquad (i=1,2,\ldots,m)$

In all, eight additional constraint equations must be added for each student to properly formulate the simple example as an integer program. This explosion of constrain equations is typical for integer programming problems and is prohibitive in formulating realistic problems.

4.13 - SET COVERING/ COMPLETE ENUMERATION

An alternative approach to solving the Level 2 problem is through complete enumeration. By listing all possible combinations of solutions to the problem the best feasible combination can be selected. This process generates thousands of possible combinations for even the simplest problem and requires a large amount of processing time. Improvements can be made by quickly eliminating obviously unreasonable solutions once a "good" solution is found. The major airline industry uses this enumeration and elimination process in solving aircraft and crew scheduling. According to Simpson 55 it is not unlikely for an airline to have several million alternatives available when searching for the optimal solution. Due to the amount of computer power required to solve problems through complete enumeration, it is impractical to use this technique for the Level 2 problem.

4.14 - HEURISTICS

In many cases involving integer programming practitioners must turn to heuristics in order to "solve" real-world problems. Often the specific problem has special characteristics which lend themselves to heuristic techniques. Moreover, according to Fisher[24], even if an optimization method is applicable, the first step in such a method is usually the application of a heuristic to obtain a good starting solution.

However, several problems may arise when using the heuristic approach. First, the heuristic only provides an approximate solution to the integer program. Secondly, the heuristic method does not lend itself to sophisticated analysis of its performance. And third, the heuristic solution is usually highly problem specific and cannot be modified to solve other similar problems. Nonetheless, even though recent work by Cornuejols, Fisher and Nemhauser[18]; Geoffrion and Graves[30]; Marsten, Muller and Killion[45]; and others[42,58] provide a growing list of successful applications of optimization methods in solving real problems, there is still a large gap that only heuristic methods can currently fill.

4.2 - HEURISTIC METHOD FOR THE LEVEL 2 PROBLEM

One approach for obtaining an approximate solution to the Level 2 scheduling problem is a heuristic which utilizes a two-step simple assignment process. In the following three sections I will discuss the heuristic solution method; its advantages; its potential worst case problems, and methods to accomodate those problems.

4.21 - STEP 1

In the first step of the method the unit scheduler attempts to ensure that each student has at least one activity. In this step every effort is made to match a student with his/her assigned instructor. In short, the scheduler wishes to optimally assign each student once. In the Level 2 problem of matching m students to n tasks, $m<n$, the heuristic accomplishes this by using the network transportation formulation described in section 4.11. As was mentioned earlier, the minimum cost formulation will assign tasks to students in such a manner such that each student is assigned a task and the total weight is minimal. The first step, then, selects the best set of m assignments from all possible sets of m assignments. The solution will be optimal for the Step 1 problem and will also perform an infeasibility check. Since the formulation has a supply of 1 at each student node the solution requires that each student be scheduled once. With the minimum weight objective function the formulation will always utilize feasible arcs when possible since the weight of artificial arcs is prohibitively high. Thus, if the optimal Step 1 solution includes more than n-m arcs with $w_{ij} = M$ the solution is infeasible. This "student lock-out" infeasibility is discussed in detail in section 4.24.

4.22 - STEP 2

After optimally assigning m tasks to the student group the scheduler then attempts to assign the remaining n-m tasks to the available students. To do this using the assignment algorithm of Step 1 two updates must be made to the graph, G, of figure 4.1. The first update raises to M the weight on all arcs incident to the tasks assigned in Step 1. Thus any assignment of a student to these tasks in Step 2 is infeasible since the tasks were already assigned in the previous step. The second update involves the feasibility of remaining assignments based on the results of Step 1. Using the adjacency list the algorithm updates all remaining feasible arcs from Step 1 to determine if they are still feasible in Step 2. If student i is assigned to task j in Step 1 then, necessarily, the student is prohibited from accomplishing any remaining task which is not on the adjacency list of j. Thus, the weight of all arcs linking i to tasks which conflict with j is increased to M. After these updates, the set of arcs, *A'*, consist only of those arcs in *A* which are feasible for 1) the Instructor Preference matrix, 2) the Step 1 assignments, and 3) the Adjacency (no conflict) List. The Step 2 assignment problem with updated feasible arc set *A'* for the sample problem is in figure 4.3. The number of feasible arcs for this problem is reduced from 29 to 9. The unassigned tasks from Step 1 are in bold print.

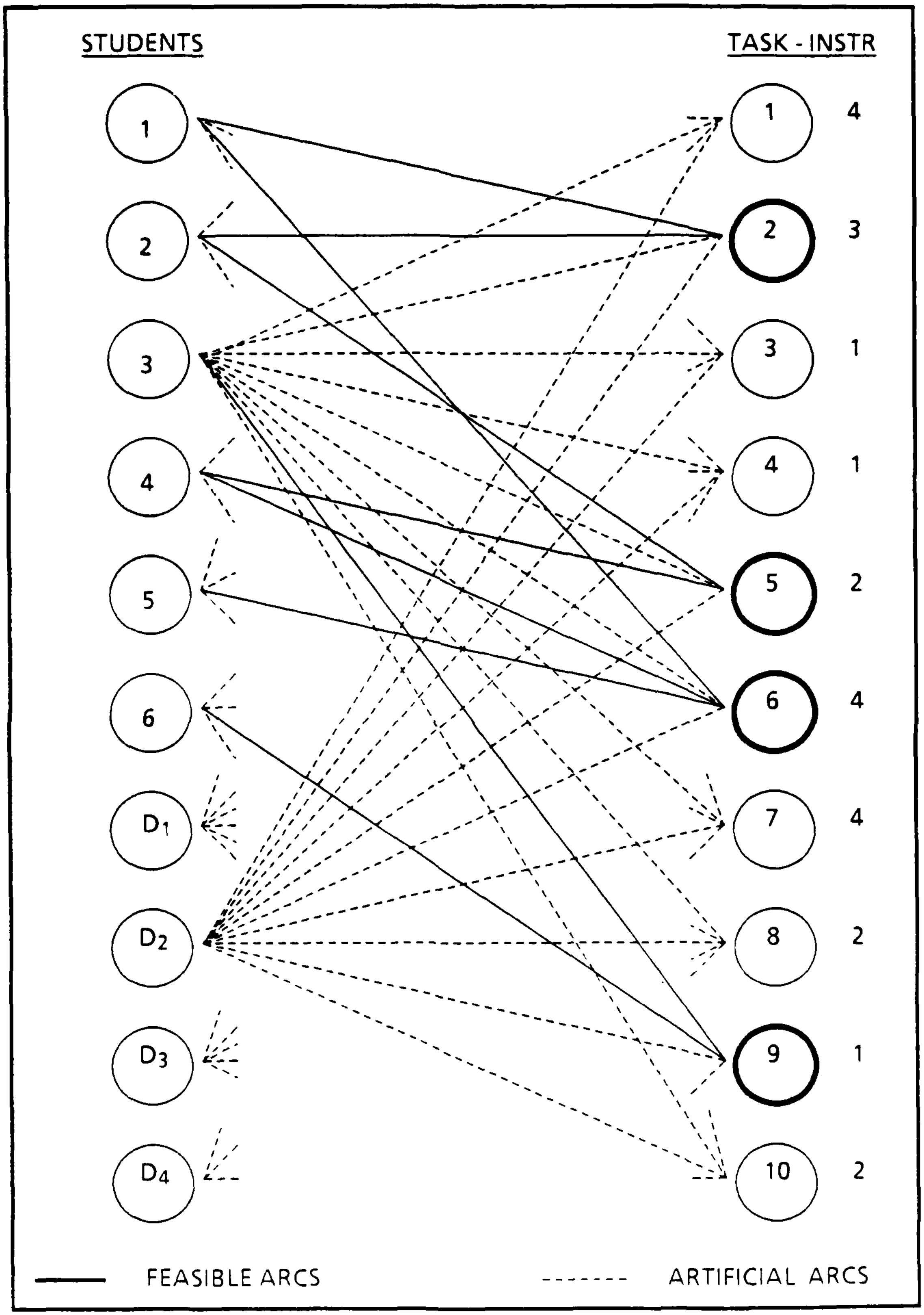


Figure 4.3 - Step 2 Sample Assignment Problem

With these updates, a second set of assignments is made using the same optimality criterion as in the previous step. In essence, after the updates, Step 2 is optimized as an independent problem. The set of assignments is optimal for the Step 2 problem. This does not imply that the combination of locally optimal solutions for the two steps is optimal for the schedule as a whole. However, if both steps have optimal solutions the combined solution to the entire schedule will be feasible and, in most cases, quite close to optimal. The combined solution for the sample problem is in figure 4.4. Notice that all tasks are assigned to students(i.e. no tasks are assigned to 'dummy' nodes), all multiple task assignments are feasible, and that 9 of the 10 matchings link the student with his/her preferred instructor.

Just as in Step 1, infeasibility can enter the problem in Step 2. Although the student lock-out infeasibility does not apply since all students are assigned a task in Step 1, it is possible to have a "task lock-out" infeasibility in this step. Task lock-out infeasibility occurs when no student is available to accomplish a specific task. This type of infeasibility is discussed in section 4.24.

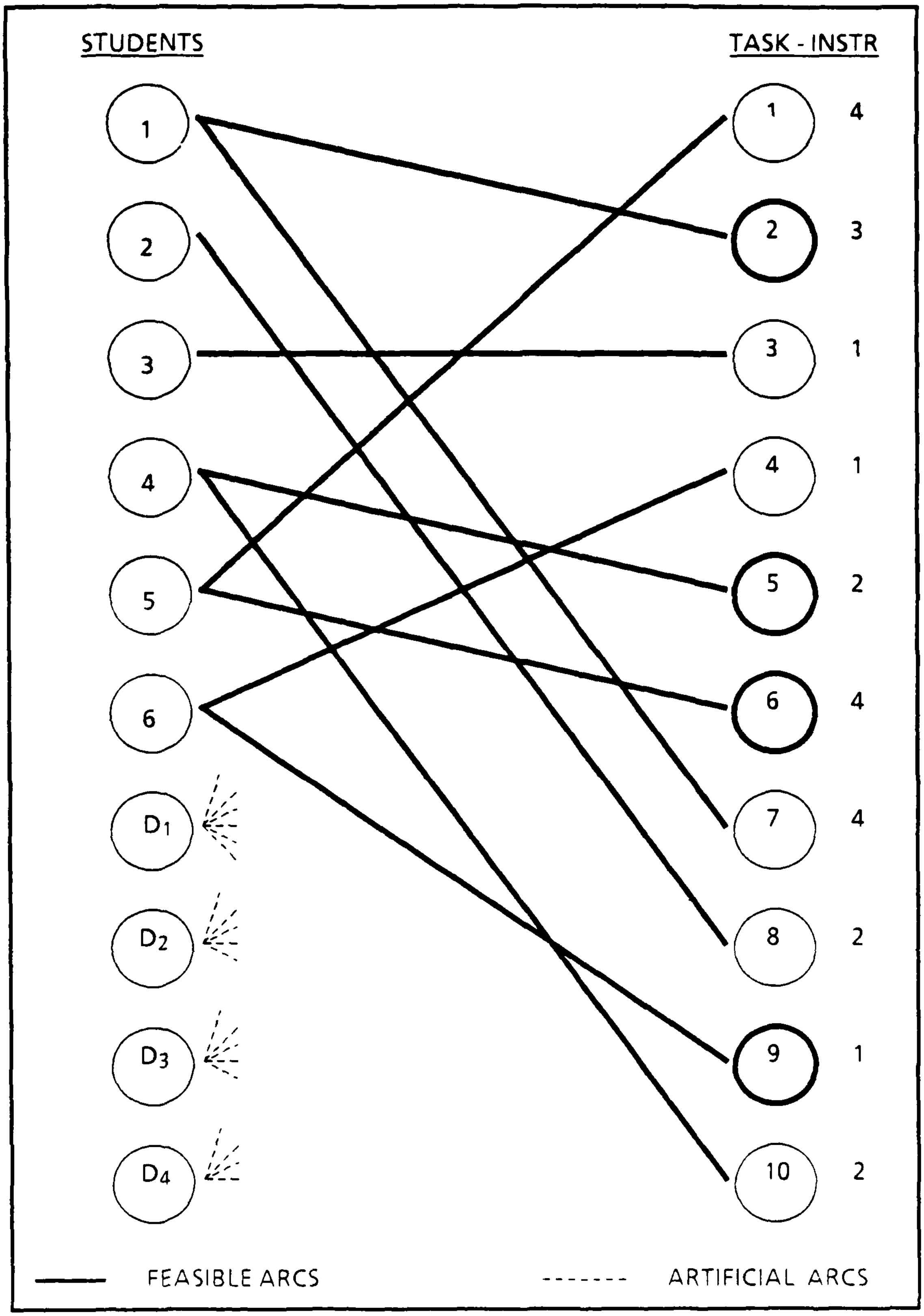


Figure 4.4 - Heuristic Solution to the Sample Assignment Problem

4.23 - ADVANTAGES TO THE HEURISTIC METHOD

There are several advantages to the heuristic solution method for the Level 2 scheduling problem.

1)The first advantage is simplicity. Because the heuristic models the problem as a two step simple assignment problem, any network flow algorithm will solve the problem directly with integer solutions. Computer programming is straightforward, and many off the shelf network flow algorithms are available.

2) Another advantage is speed. Since the solution method uses a network flow algorithm, the Level 2 solution is obtained in seconds. The speed of the entire process is limited only by the time necessary for data entry.

3) A major advantage of the heuristic is the weight updating procedure. Integer programming and other optimization methods do not allow elaborate weight updating with such simplicity as the two step heuristic. In addition to updating arcs for feasibility after the Step 1 assignments, the heuristic can update remaining feasible arcs for desirability as well. Recall that the objectives of the unit scheduler are threefold.

1. Schedule each student at least once (Step 1)
2. Match students with more preferred instructors (Step 1, 2)
3. Maximize student idle time between events (Step 2)

The third objective, previously unaddressed, can easily be accomodated by the heuristic. Based on the assignments of Step 1, the scheduler can assign lower weights to those arcs in A' which provide greater idle time between feasible tasks for each student. This arc weight modification results in Step 2 assignments which favor greater idle time for the students involved.

4) A fourth advantage is acceptability. The heuristic method employed to solve the Level 2 problem closely models actual scheduling practices used

by unit schedulers on a daily basis. Rarely does a unit scheduler have the time to optimally assign the entire schedule by hand. His/her realistic goal is to provide each student with one "good" instructor-task matching and to merely 'fill' the remaining tasks with feasible pairings. The two step assignment method accomplishes these goals easily in much the same manner as the unit scheduler. This will aid in the method's acceptability to current unit schedulers since it does not require radical changes in scheduling practices.

5) Finally, the Level 2 heuristic is an optimization-based heuristic. Optimization methods are an integral part in the design of the heuristic. This master-slave relationship is very effective in solving many real world problems which do not lend themselves to optimization techniques directly. Ball et al.[5] demonstrates the advantage of using matching algorithms as an ingredient of the heuristic for vehicle scheduling.

4.24 - PROBLEMS WITH THE HEURISTIC METHOD

As was mentioned in sections 4.21 and 4.22 two types of infeasibility, 'student lock-out' and 'task lock-out', can occur in the Level 2 problem. The student lock-out infeasibility, which occurs in the Step 1 process, results in an unassigned student. Figure 4.5 shows a simple example of this type of infeasibility. If a single instructor is the only member of the student preference list for 's' students and he/she performs 't' tasks, t<s, in the unit schedule then s-t students will be locked-out of the schedule. This student lock-out infeasibility is unrelated to the solution method employed and can only be avoided through proper planning by the unit scheduler. The scheduler must develop the preference list so instructors are assigned to students in

proportion to the availability of the instructor. Thus, a supervisor who is limited to one student activity per day should not be a primary instructor to more than one student. Also, those students who are assigned to supervisors should have sufficient depth in the preference list to ensure enough options exist in case the supervisor is unavailable. The sample problem is an example of proper planning to avoid this type of infeasibility. In table 4.1 instructor #3 is the supervisor and is assigned a single task by the unit scheduler. Notice in table 4.2 that no student has instructor #3 as a primary instructor and two students(1,2) have the supervisor as an alternate instructor. This scheduling foresight results in greater flexibility in the schedule and virtually eliminates student lock-out infeasibility.

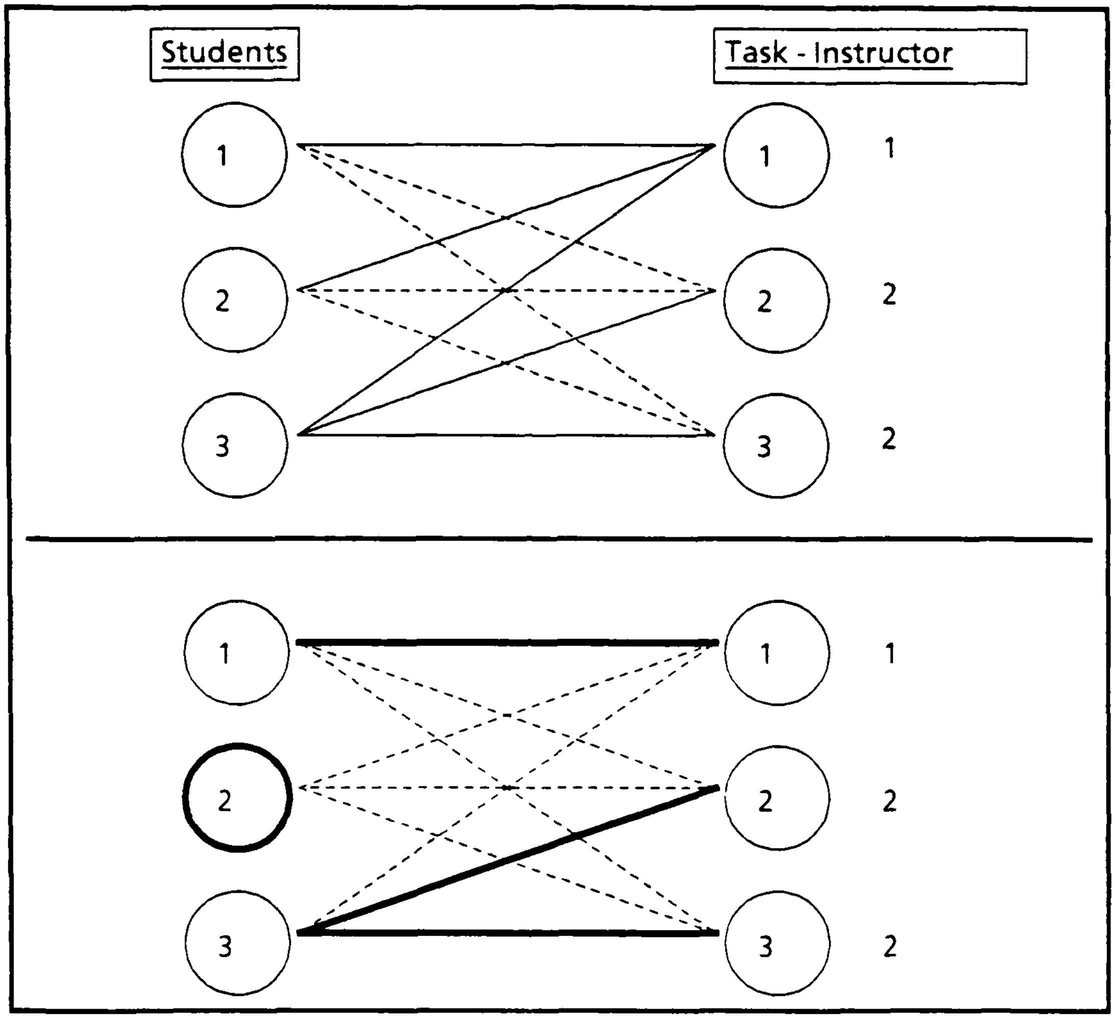


Figure 4.5 - Student Lock-out Infeasibility

The second type of infeasibility, task lock-out, occurs in Step 2 of the heuristic and results in an unassigned task. Task lock-out results primarily from the heuristic method although it can occur in optimization methods as well. However, the two step heuristic method may result in an infeasible solution where an optimization method would not. Figure 4.6 provides an example where the heuristic solution is infeasible while an optimization solution is feasible. In the example only tasks 1 and 3 can be performed by a single student. The heuristic method in 4.6b incorrectly assigns task 1 to student 2 in Step 1 because of its low cost. Unfortunately, this assignment results in infeasibility in Step 2 since no student is available for task 3. Thus, task 3 is improperly locked-out of the solution. The correct solution in 4.6c optimally assigns all tasks to students at the lowest overall cost.

Because the Step 1 assignment process does not consider its effect on the Step 2 process, task lock-out infeasibility may result. Although the problem is a serious one, for the vast majority of cases it is highly unlikely to occur. This is due to the density of feasible arcs, A, in the graph. Three factors contribute to the density. First, the student preference list usually contains two or more instructors for each student. Thus, students will have feasible arcs to several instructor-task nodes in the unit schedule.

Second, the instructor force is small relative to the students. The average instructor is accomplishing several tasks in the daily schedule. This has a multipicative effect on the number of feasible arcs for each student in the graph. Also, the large student/instructor ratio results in a low task/student ratio in the schedule. The task/student ratio usually hovers near 1.5 which indicates only 50% of the student pool is needed to solve the Step 2 problem.

Third, proper construction of the student preference lists adds to the density of A. Just as in the case of student lock-out, the scheduler can avoid task lock-out in Step 2 by properly assigning instructors to students. Task lock-out would occur if an instructor who is not on several student's lists is performing many tasks in the schedule. Thus, although students are available, none match properly with the instructor of the task. By assigning those instructors who will be doing most of the work to the majority of students sufficient density in A will occur to prevent task lock-out infeasibility.

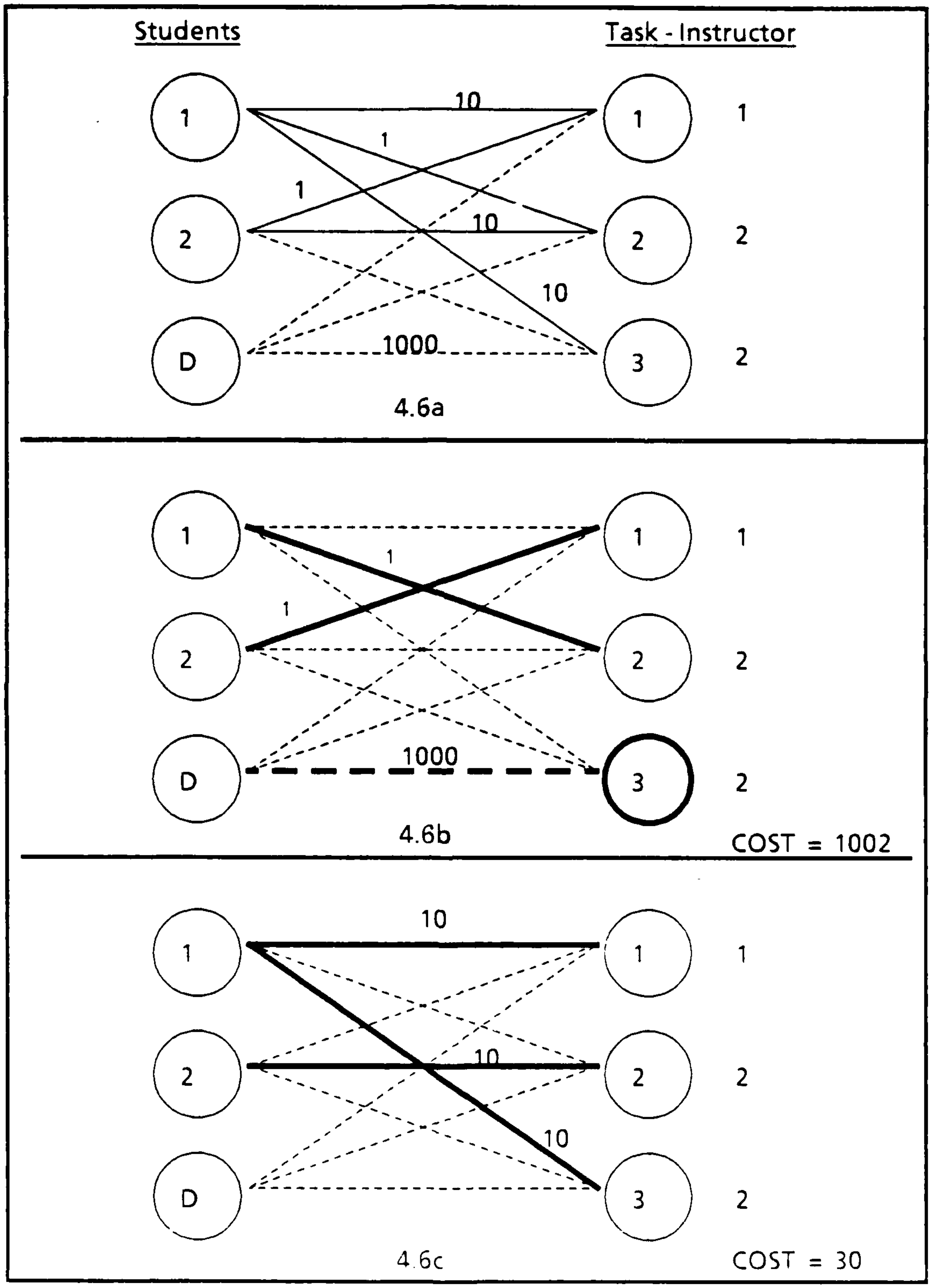


Figure 4.6 - Task Lock-out Infeasibility

The sample problem illustrates these three points well. Three instructors(1,2,4) are performing three tasks each. Also, these instructors fill all of the primary assignments and one-half of the secondary assignments in the student preference list. Thus, even though two students(3,5) have only one assigned instructor each and the task/student ratio is a high 1.67, the heuristic method solves the problem with no infeasibility. A lower task/student ratio and greater depth in the student preference list will make the possibility of task lock-out infeasibility due to the heuristic method even more remote.

CHAPTER 5

COMPUTATIONAL RESULTS

5.0 - INTRODUCTION

After sucessfully applying the algorithms of chapters 3 and 4 to the sample problem, the next logical step is to test them on actual USAF/ATC scheduling problems. The 559th Flying Training Squadron located at Randolph AFB, San Antonio, Texas supplied the test data which contains information about tasks, instructors, and students for the week of February 24-28, 1986. This data is included in appendicies A, B, and C. Unfortunately, due to the dynamic nature of the 559th scheduling environment, the number of schedulers involved, and the inability to work side by side; no direct comparison is possible between the algorithms of this paper and the actual scheduling results used by the 559th. Hopefully, greater interaction will result as an outgrowth of this paper and more conclusive analysis of the scheduling methods used may be conducted.

This chapter is divided into two sections. In the first section I will briefly discuss the Level 1 program and describe the computational results of the solution method. The second section focuses on the Level 2 program and its results. A more detailed discussion of both programs is in appendix D. All programs are in Fortran 77 and were written, compiled, and executed on an IBM PC/XT.

5.1 - LEVEL 1 PROGRAM RESULTS

Recall that the Level 1 problem is twofold. The first problem is to determine the minimum number of instructors needed to accomplished all scheduled tasks. Then the most efficient way of using the instructors is found. The Level 1 program uses the minimum cost network flow code RELAX developed at M.I.T. by Bertsekas and Tseng[9,10]. The program uses RELAX to solve both phases of the Level 1 scheduling problem.

Phase 1 of the program recasts RELAX as a maximum flow code to determine the lower bound on the number of instructors as outlined in section 3.6. The program then uses the phase 1 output as input to phase 2. Phase 2 uses RELAX to solve the minimum-cost flow problem described in section 3.7. Finally, a simple heuristic checks the task string length for each instructor and trims those strings longer than three tasks.

Table 5.1 is a compilation of the 559th task data in appendix A. The table displays the task breakdown, by type, for each day of the week. Aircraft tasks require 3 hours to accomplish, simulator tasks 2 hours 15 minutes, cross country tasks are all day affairs, and SOF/RSU/Other tasks are variable in length.

Table 5.1 - Task breakdown by type and day

TASK TYPE	MON.	TUES.	WED.	THURS.	FRI.
Aircraft	50	55	57	55	49
Simulator	28	26	30	25	30
Cross Country/O&B	5	6	5	4	13
RSU/SOF/Other	7	8	7	7	7
TOTAL	90	95	99	91	99

Table 5.2 displays the phase 1 and phase 2 output on the number of instructors needed to fill the daily schedules. On average, the program determined the lower bound(phase 1 output) in approximately 20 seconds with the longest running under 30 seconds. The phase 2 numbers also required less than 30 seconds. Row 3 shows the difference in the output. This difference is a function of the length-of-day and task-chain length restrictions imposed on the Level 1 problem. Recall the phase 1 output is the solution to the relaxed problem without these constraints. In phase 2 the program attempts to match the phase 1 output but adds an instructor for every constraint violation in the relaxed solution. Thus, the phase 2 schedule output will always be feasible for the number of instructors listed by the phase 2 program.

Row 4 of table 5.2 displays the percentage of violations in the phase 1 solution with respect to the total number of tasks scheduled. The average percentage of violations for the week is 5.3%. Thus, the relaxed solution output of phase 1 is, on average, 95% accurate and requires only minor adjustments by the heuristic. Based on this test, the percentage of violations appears to decrease on those days with greater number of tasks. Only the phase 2 output for Monday is significantly different than the other days.

However, there is nothing unique about the task breakdown or schedule makeup for that day which would indicate a high percentage of violations in the phase 1 solution. The deviation may be due to the low total number of tasks and low aircraft/simulator ratio on Monday. Excluding the output for Monday, the average percentage of violations drops to 4.2% for the remaining four days.

Table 5.2 - Number of Instructors Needed

PHASE	MON.	TUES.	WED.	THURS.	FRI.
PHASE 1 (Lower Bound)	31	38	37	34	42
PHASE 2 (# In Schedule)	40	43	41	38	45
PHASE 2 - PHASE 1	9	5	4	4	3
% Violation	10.0 %	5.3%	4.0%	4.4%	3.0%

Figures 5.1 is a summary of the daily phase 2 schedules generated by the Level 1 program. The average total elapsed time from program initiation to schedule printout was less than 90 seconds with little variance. The output contains feasible task strings for the instructors and flags all duty day violations. The heuristic program prints "Duty day violation" and adds an instructor for al task strings with duration longer than 12 hours. Only four such violations exist for the entire week. The vast majority of singleton task strings in the schedules are tasks of long duration such as cross country missions or SOF duties. Two and three task strings are made up primarily of the shorter duration aircraft, simulator, and RSU tasks.

TASK STRING NUMBER	PHASE 2 DAILY SCHEDULES				
	MON.	TUES.	WED.	THURS.	FRI.
1	79	65	88	79	80
2	80	66	89	81	81
3	81	67	90	82	82
4	82	82	91	83	83
5	83	83	92	84	84
6	1-19-89	84	94	86	85
7	2-20-38	85	1-21-98	90	86
8	3-21-39	86	2-23-46	1-18-37	87
9	4-88-34	87	3-24-47	2-19-38	88
10	76	89	4-25-48	3-20-39	89
11	5-22-40	93	5-27-50	4-21-75	90
12	6-24-36	1-23-46	6-28-51	5-22-41	91
13	78	2-24-47	7-29-52	6-23-42	92
14	7-23-41	3-25-48	8-30-53	7-67-73	1-20-98
15	8-26-44	4-26-49	9-73-80	52	2-21-41
16	9-25-37	5-27-50	10-35-81	8-25-44	3-22-42
17	10-28-86	6-28-51	11-32-55	9-26-45	4-23-43
18	D. D. V.*	7-29-52	12-33-56	10-27-46	5-60-68
19	11-29-42	8-30-53	13-36-85	11-29-48	6-24-44
20	12-27-45	9-31-54	58	12-30-49	7-26-40
21	13-31-49	10-32-55	14-37-86	13-31-50	8-27-47
22	14-32-50	11-33-70	15-38-87	14-32-87	9-29-49
23	15-33-75	12-35-72	59	15-33-53	10-30-72
24	16-66-74	13-36-73	16-39-95	16-34-54	11-31-74
25	17-35-77	56	60	17-36-80	12-64-73
26	18-30-43	14-37-74	17-40-99	56	13-32-75
27	51	15-38-75	61-18-41	57-65-35	14-33-76

Figure 5.1 - Phase 2 Schedules for the Week

TASK STRING NUMBER	PHASE 2 DAILY SCHEDULES				
	MON.	TUES.	WED.	THURS.	FRI.
28	52-57-64	57	62-26-49	55	15-35-78
29	72	16-39-76	63-97-44	58-71-78	16-36-79
30	53-59-67	17-40-77	64-70-77	59-64-51	17-34-77
31	46	58	65-71-78	60-28-47	50
32	54-60-68	18-41-78	66-31-54	61-68-40	51-19-39
33	47	19-42-79	67-74-82	62-69-76	52-59-67
34	55-61-69	59	68-75-83	63-70-77	99
35	48	20-43-80	69-76-84	66-72-91	53-61-69
36	56-63-71	88	93-34-57	88	54-28-48
37	58-65-73	21-44-81	D. D. V.*	85-24-43	56-63-71
38	84-62-70	60-34-71	96-72-79	89-74	57-62-70
39	87-85-90	61-69-95	19-42		58-66-46
40	D. D. V.*	64-94-91	20-43		93-65-95
41		92-22-45	22-45		D. D. V.*
42		62-69			96-25-45
43		63-90			18-38
44					55-94
45					97-37

Figure 5.1 - Phase 2 Schedules for the Week

5.11 - SCHEDULE ADJUSTMENTS

If the number of instructors planned for by the squadron scheduler is less than the number required to fill the phase 2 schedule in row 2 of table 5.2 above, the scheduler has two options. The scheduler can add instructors to the schedule, if available and rerun the program to obtain an improved feasible schedule. If no additional instructors are available, the current number of tasks is infeasible and the scheduler must eliminate low priority tasks from the schedule. For example, in the schedule for Monday (see figure 5.1), if the 1655 simulator task (#78) in bold print in line 13 is a low priority task, the scheduler can easily eliminate it and reduce by one the required number of instructors.

Conversely, if the number of instructors required by the phase 2 schedule is less than the number available, the scheduler can add tasks to the schedule(meetings, appointments, etc.) or relieve the extra instructors from duties for the day.

After the squadron scheduler makes minor adjustments to the phase 2 schedule to best suit the squadron's needs, he/she then divides the task strings among the six units in the squadron. This decomposition is based on the "contracts", mentioned in chapter 1, submitted by each unit scheduler. The actual unit contracts for the week are in appendix B. From these contracts the squadron scheduler knows the desired task makeup of each unit for every day of the week. For example unit "B" requested the following sixteen tasks for its seven instructors for Wednesday, Feb. 26th:

1) 1 SOF task
2) 1 RSU task
3) 8 Aircraft missions
4) 6 Simulator missions

Table 5.3 shows the contracted work schedule, broken down by instructor, for unit B. Instructor #1 is the supervisor and must perform the SOF task while instructor #6 is qualified to perform the RSU task. All instructors can perform any combination of aircraft or simulator tasks, but no instructor may have more than three tasks. Finally, unit B is to work the early shift in the squadron.

Table 5.3 - Unit B Contract, by Instructor

INSTRUCTOR NUMBER	TASK PERIOD		
	1	2	3
1	OFF	OFF	SOF
2	OFF	1	OFF
3	1	1	1
4	1	1	1
5	1	1	1
6	1	RSU	1
7	1	1	OFF
TOTAL = 16	5	6	5

Using the phase 2 schedule for Wednesday, unit B's contract, and knowing unit B is on the early shift: the squadron scheduler could select the task assignments listed in table 5.4. In all, six task strings are used to fill the contract for unit B exactly with all the items requested above. One task string is divided between instructors #2 and #7.

Table 5.4 - Assigned Schedule for Unit B , by Instructor

INSTRUCTOR NUMBER	TASK-STRING LINE NUMBER (from fig. 5.1)	TASK PERIOD		
		1	2	3
1	6	OFF	OFF	94 (SOF)
2	15	OFF	80 (SIM)	OFF
3	28	62 (SIM)	26 (ACFT)	49 (ACFT)
4	32	66 (SIM)	31 (ACFT)	54 (ACFT)
5	27	61 (SIM)	18 (ACFT)	41 (ACFT)
6	29	63 (SIM)	97 (RSU)	44 (ACFT)
7	15	9 (ACFT)	73 (SIM)	OFF
TOTAL = 16		5	6	5

The squadron scheduler would repeat this process for the remaining five units until all 99 tasks in the Wednesday schedule are filled.

5.2 - LEVEL 2 PROGRAM RESULTS

The Level 2 problem discussed in chapter 4 addresses the student assignment process at the unit level. As with the Level 1 program, the Level 2 optimization-based heuristic program uses the minimum-cost network flow code RELAX in both steps of the program.

Both Step 1 and Step 2 of the heuristic program recast RELAX as a transportation network code to solve the student assignment problems described in section 4.2. For each step RELAX required less than 3 seconds to determine the optimal student-task pairings for the test problems. This speed is primarily a function of smaller problem size encountered by the unit scheduler.

For the Level 2 test I will continue to use the unit B data for Wednesday, February 26th. The Level 1/Phase 2 schedule for the instructors of unit B, summarized in table 5.4, is the input to the two step Level 2 program. Additional data required by the Level 2 program is the Student Preference Matrix for the unit B students. The actual preference matrix from appendix C for the ten students in unit B is displayed in table 5.5.

Table 5.5 - Student Preference matrix

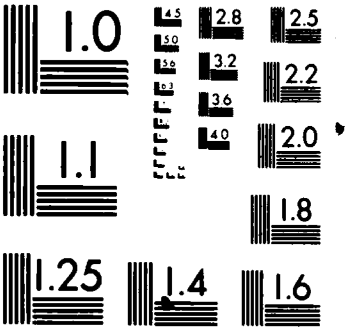
STUDENT NUMBER	INSTRUCTOR NUM.			
	1st	2st	3rd	4th
1	2	7	3	
2	6	3		
3	5	6		
4	7	6	1	2
5	2	1		
6	4	3	5	
7	7	3		
8	5	3		
9	6	2		
10	4	5		

Notice in the table the effective assignment of instructors to students accomplished by the unit B scheduler. Instructor #1, the supervisor, is not a most preferred instructor for any student. Also, the supervisor is only included in the lists for two students (4,5). The remaining 23 entries in the matrix are evenly divided both in number and order of preference among the six remaining unit B instructors. This effective planning results in no student or instructor being dominant in the assignment process, increases scheduling flexibility, and greatly reduces the likelihood of encountering student or task lock-out infeasibility in the solution. The task/student ratio is a reasonable 1.6. This ,too, reduces the possibility of infeasibility.

Table 5.6 contains the results of the Level 2 program for unit B. This schedule displays the task number, instructor and student for each task, and a brief description of each task. The time required from Level 2 program initiation to printed output was less than 15 seconds. No task or student lock-out infeasibility occured as each task is assigned to a student, and each student performs at least one activity. Of the 16 assignments, 12 match a student with his/her 1st preference instructor, 3 match the student with the 2nd preference instructor, and 1 requires a student to perform a task with a 3rd preference instructor. The task assignments for the six students performing two tasks each (1,2,3,7,9,10) are all feasible and have an average idle time of 2.16 hours between tasks.

AD-A172 321 AIDING USAF/UPT (UNDERGRADUATE PILOT TRAINING) AIRCREW SCHEDULING USING NETWORK FLOW MODELS(U) AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH R M HUELSKAMP JUN 86 2/2

UNCLASSIFIED AFIT/CI/NR-86-90T F/G 5/9 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Notice in the table the effective assignment of instructors to students accomplished by the unit B scheduler. Instructor #1, the supervisor, is not a most preferred instructor for any student. Also, the supervisor is only included in the lists for two students (4,5). The remaining 23 entries in the matrix are evenly divided both in number and order of preference among the six remaining unit B instructors. This effective planning results in no student or instructor being dominant in the assignment process, increases scheduling flexibility, and greatly reduces the likelihood of encountering student or task lock-out infeasibility in the solution. The task/student ratio is a reasonable 1.6. This ,too, reduces the possibility of infeasibility.

Table 5.6 contains the results of the Level 2 program for unit B. This schedule displays the task number, instructor and student for each task, and a brief description of each task. The time required from Level 2 program initiation to printed output was less than 15 seconds. No task or student lock-out infeasibility occured as each task is assigned to a student, and each student performs at least one activity. Of the 16 assignments, 12 match a student with his/her 1st preference instructor, 3 match the student with the 2nd preference instructor, and 1 requires a student to perform a task with a 3rd preference instructor. The task assignments for the six students performing two tasks each (1,2,3,7,9,10) are all feasible and have an average idle time of 2.16 hours between tasks.

CHAPTER 6

CONCLUSIONS AND EXTENSIONS

6.0 - THESIS REVIEW

Undergraduate Pilot Training is a dynamic, high pressure program which produces front line aviators for the United States Air Force. The eleven month curriculum contains a complicated assortment of academic, flight, and simulator training. Scheduling the instructors, students, and activities on a daily basis is the responsibility of the squadron scheduler and six unit schedulers. For many years the manual method of scheduling currently used by the schedulers was adequate due to the nature of training. Small class sizes and more dependable aircraft systems resulted in fewer scheduling constraints. In recent years, however, the need for optimization based scheduling has become paramount. Class sizes have increased dramatically, and the aging aircraft fleet has become less reliable. As constraints on the scheduling process grow even tighter, feasible solutions to daily schedules become more elusive. The current scheduling process consumes many man-hours as the manual heuristic becomes less effective.

In this paper we attempt to alleviate some of the scheduling difficulties by developing a micro computer based algorithm which will quickly find feasible solutions to both the squadron and unit level scheduling problems. In the past, micro computers lacked the necessary capacity to slove the UPT scheduling problem, and larger mini or mainframe computers were not available for use by the UPT squadrons. However, with the advent of powerful micro computers and by dividing the scheduling process into two smaller problems we obtain tractable problems which can be solved on a micro computer.

In chapter 3 we address the instructor scheduling problem and formulate it as a network flow problem. Using a maximum flow formulation and Dilworth's theorem we find the minimum number of instructors needed to accomplish all tasks in the schedule. Then, using the minimum cost flow formulation we tightly schedule tasks to instructors in order to best utilize the instructor force.

With the Level 1 output of chapter 3 we then decompose the problem into the six smaller units. The Level 2 optimization-based heuristic is chapter 4 assigns students to tasks based on the previous Level 1 instructor assignments. We formulate the student scheduling problem as an assignment problem. The heuristic utilizes a 2 step network transportation formulation to feasibly assign all tasks to students. With this result the scheduling process is complete. The Level 1 and Level 2 programs assign tasks to instructors and students in a manner which accomodates each individual's needs and preferences.

6.1 - CONCLUSIONS

As a result of this research effort we have made several conclusions about the current UPT aircrew scheduling efforts and the solution method proposed in this paper.

1) Our solution method is much faster. Currently, seven to ten squadron officers expend up to 100 man-hours daily on the UPT scheduling problem. The solution methods in this paper require only a few minutes on a micro computer to generate feasible schedules for the entire squadron.

2) Our solution method is less expensive. The current scheduling method not only requires a great amount of time, it also demands an extremely high skill level to implement. Thus, the squadron must use experienced instructors to construct daily schedules. The micro computer based method in this paper requires a much lower skill level for day-to-day operation. The skill of the experienced scheduler is captured by the various cost functions, task priorities, and preference matricies. Thus, fewer and less skilled people can effectively complete the daily schedule, and highly skilled officers are available for more important instructional duties.

3) Our solution method results in better schedules. Due to the complexity of the scheduling problem and the inadequicies listed above, the manual method often results in ineffective or infeasible use of resources. The optimization based solution method in this paper exploits the underlying network structure of the problem, performs a fast infeasibility check on the problem, and always produces feasible schedules. Also, the method schedules instructors tightly to eliminate long duty days. Thus, fewer instructors can accomplish the daily schedule in less time.

4) Our solution method is interactive. Studies have shown that the most successful computer based scheduling packages are those which are a tool, not a replacement, for the master scheduler. The interactive nature of our solution method adds flexibility by providing an expert scheduler the ability to modify the schedule to best fit the squadron's needs. The interactive approach also greatly simplifies the programs as complicated or "fuzzy" decisions are left to the master scheduler.

Thus, the union of a master scheduler and an effective, optimization based solution method results in a low cost scheduling process. The interactive process produces feasible schedules in minutes and releases highly skilled instructors to perform more important tasks.

6.3 - EXTENSIONS

There are several areas which one can explore as an extension to this thesis. Preferably, any future research on the UPT scheduling problem can be sponsored by the USAF Air Training Command and involve a high level of interaction between the researcher and the practitioner. For this reason future research efforts would be most effective if conducted in conjunction with Air Training Command headquarters in San Antonio, Texas.

The first extension is to analyze the effects of different cost functions on the solution method. The use of non-linear or quadratic cost functions may more accurately reflect the desired practices of the squadron. This would enhance the acceptability of the solution method by providing more realistic schedules. Again, without extensive interaction with actual UPT schedulers elaborate cost function updates may be largely ineffective.

A second extension is to further analyze the phase 2 performance of the Level 2 method. The worst case performance of the Level 2 heuristic is improper infeasibility and can result from a number reasons. Thus, further investigation of the robustness of the heuristic is possible. Variables affecting the heuristic method are 1) the total number of tasks, instructors, and students in the schedule, 2) the task/student ratio, and 3) the depth of the student preference lists. Analysis of the heuristic method in these areas would determine the usefulness of the method on larger and more complex problems.

Investigating the applicability of exact algorithms for the Level 2 scheduling problem is a third extension of this thesis. Although we mention in chapter 4 that exact algorithms currently are not executable for the UPT problem on micro computers, the power and capabilities of these computers is rapidly advancing. Thus, one could investigate the applicability of set covering, integer programming, branch and bound, and/or Lagrangian relaxation methods to better solve the Level 2 problem.

Finally, the UPT scheduling problem is one which is amenable to the field of Expert Systems. Expert systems work best on those problems which have simple structure but require a large degree of expertise to solve. The dynamic nature of the UPT scheduling environment demands experienced squadron and unit schedulers who must adapt student and instructor assignments to changing weather and revised task schedules. The mental checklists used by the chedulers would be a basis for the expert system which could then be used by less skilled workers.

REFERENCES

1. E. Baker, *Efficient Heuristic Solutions for the Airline Crew Scheduling Problem*. D.B.A. Thesis, University of Maryland, College Park, Maryland (1979).

2. E. Baker, L. Bodin and M. Fisher, *The Development and Implementation of a Heuristic Set Covering Based System for Air Crew Scheduling*. Working paper #80-015, University of Maryland(December 1980).

3. E. Balas and M. Padberg, "Set Partitioning: A Survey", *SIAM Review*, **18**(4), 710-761 (1976).

4. M. Ball, *A Comparison of Relaxations and Heuristics for Certain Crew and Vehicle Scheduling Problems*. Presented at National ORSA/TIMS Meeting, Washington, D.C.(1980).

5. M. Ball, L. Bodin and R. Dial, "Experimentation with a Computerized System for Scheduling Mass Transit Vehicles and Crews". *Comp. Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, 313-336. North Holland, Amsterdam (1981).

6. R. Baron and L. Shapiro, *Data Structures and Their Implementation*. Van Nostrand Reinhold Co., 56-68, (1980).

7. T. Bartlett, "An Algorithm for the Minimum Number of Transport Units to Maintain a Fixed Schedule". *Naval Res. Logistics Quarterly*, 4, 139-149 (1957).

8. G. Bennington and K. Rebibo, "Overview of RUCUS Vehicle Scheduling Program (BLOCKS)". *Workshop on Automated Techniques for Scheduling of Vehicle Operators for Urban Public Transportation Services* (Edited by D. Bergmann and L. Bodin) (1975).

9. D. Bertsekas and P. Tseng, *Relaxation Methods for Minimum Cost Ordinary and Generalized Network Flow Problems*. Laboratory for Information and Decision Systems Report LIDS-P-1462, M.I.T. (May 1985).

10. D. Bertsekas and P. Tseng, *Relax: A Computer Code for the Minimum Cost Network Flow Problems*, Laboratory for Information and Decision Systems Report LIDS-P-1469, M.I.T. (May 1985).

11. L. Bodin and R. Dial, "Hierarchical Procedures for Determining Vehicle and Crew Requirements for Mass Transit Systems". *Transportation Resources Record*, **746**, 58-64 (1980).

12. L. Bodin and B. Golden, "Classification in Vehicle Routing and Scheduling". *Networks* 11(2), 97-108 (1981).

13. L. Bodin, B. Golden, A. Assad and M. Ball, "Routing and Scheduling of Vehicle and Crews: The State of the Art". *J. Computers and Operations Research*, 10(2), (1983).

14. L. Bodin, D. Rosenfield and A. Kydes, "UCOST, A Micro Approach to a Transit Planning Problem". *Journal of Urban Analysis* 5(1), 47-69 (1978).

15. S. Bradley, A. Hax and T. Magnanti, *Applied Mathematical Programming*, Addison-Wesley Publishing Co. Reading, Mass. 310-402, (1977).

16. C. Carpenter, "Computer Use in Nursing Management". *J. of Nursing Admin.* , 17-21, (Nov. 1983).

17. G. Clarke and J.Wright, "Scheduling of Vehicles From a Central Depot to a Number of Delivery Points". *Ops. Res* **12**, 568-581 (1964).

18. G. Cornuejols, M. Fisher and G. Nemhauser, "Location of Bank Accounts to Optimize Float: an analytic study of exact and approximate algorithms". *Management Sci.*, **23**, 789-810 (1977).

19. G. Dantzig and D. Fulkerson, "Minimizing the Number of Tankers to Meet a Fixed Schedule". *Naval Res. Logistics Quart.* **1**, 217-222 (1954).

20. M. Davis, "A Resource Assignment and Management Information System for Event Scheduling in a Flight Training Environment". *Interfaces* **10**(4), 68-74 (1980).

21. J. Edmonds and R.M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems". *Journal of the Assn. of Computing Machinery*, **19**, 248-264 (1972).

22. S. Even and R. Tarjan, "Network Flow and Testing Graph Connectivity". *J. SIAM Comp.*,4(4), 507-512 (1975).

23. W. Finnegan, "A Network Model for Bidline Generation". *FEC Technical Report*, Federal Express Corporation, Memphis, Tenn. (1977).

24. M. Fisher, "Worst-case Analysis of Heuristic Algorithms". *Management Sci.* **26**(1), 1-17 (January 1980).

25. M. Fisher, "The Lagrangian Relaxation Method for Solving Integer Programming Problems". *Management Sci.* **27**(1), 1-17 (1980).

26. M. Fisher and R. Jaikumar, "An Algorithm for the Space-Shuttle Scheduling Problem". Ops. Res. **26**(1), 166-182 (1978).

27. M. Fisher, R. Jaikumar, L. N. Van-Wassenhove, *A Multiplier Adjustment Method for the Generalized Assignment Problem*, unpublished, University of Penn. (1981).

28. L. Ford and D. Fulkerson, *Flows in Networks* Princeton University Press, Princeton, N.J. (1962).

29. H. Gabbay, *An Overview of Vehicular Scheduling Problems* M.I.T. Operations Research Center Technical Report # 103 (September 1974)

30. A. Geoffrion and G. Graves, "Multicommodity Distribution System Design by Benders Decomposition". *Management Sci.* **20**, 822-844 (1974).

31. I. Gertsbach and S. Helman, "Minimal Resources for Fixed and Variable Job Schedules". *Ops. Res.* **26**(1), 68-85 (1978).

32. B. Gillett and L. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem". *Ops. Res.* **22**, 340-349 (1974).

33. F. Glover, R. Glover and C. McMillan, "The General Employee Scheduling Problem: An Implemented, Heuristic Solution System". *Management Science Report Series* **84-1**, (1984).

34. V. Godin, "Interactive Scheduling: Historical Survey and State of the Art". *AIEE Transactions*, **10**(3), 331-337 (1978).

35. R. Graham, E. Lawler, J. Lenstra and A. Rinnooy Kan, "Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey". *Ann. Discrete Math.* **5**, 287-326 (1979).

36. S. Graves, "A Review of Production Scheduling". *Ops. Res.* **29**(4), 646-675 (1981).

37. S. Graves and J. Shapiro. *Problem Formulations and Numerical Analysis in Integer Programming and Combinatorial Optimization*. Working paper, Operations Research Center OR-095-79, M.I.T. (1979).

38. J. Hopcroft and R. Karp, "A $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Grtaphs", *J. SIAM Comp.* 4(4), 507-512, (1975).

39. H. Kuhn, "The Hungarian Method for the Assignment Problem". *Naval. Res. Logistics Quart.*, **2**, 83-97, (1955).

40. E. L. Lawler, *Combinatorial Optimization: Networks and Matroids*, Holt, Reinhart, and Winston, New York, New York (1976).

41. J. Lenstra and A. Rinnooy Kan, "Complexity of Vehicle Routing and Scheduling Problems". *Networks* 11(2), 221-227 (1981).

42. J. Leung, *Polyhedral Structure of Fixed Charge Problems and a Problem in Route Planning*, PhD. Thesis, M.I.T (1985).

43. T. Magnanti, "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects". *Networks* 11(2), 179-214 (1981).

44. T. Magnanti and R. Wong, "Network Design and Transportation Planning: Models and Algorithms". *Trans. Sci.* **18**(1), 1-55 (1984).

45. R. Marsten, M. Muller and C. Killion, "Crew Planning at Flying Tiger: A Successful Application of Integer Programming". *Management Sci.* **25**(12), 1175-1183 (1979).

46. R. Marsten and F. Shepardson, "Exact Solution of Crew Scheduling Problems Using the Set Partitioning Model: Recent Successful Applications". *Networks* 11(2), 167-177 (1981).

47. C. Monma and M. Segal, "A Primal Algorithm for Finding Minimum-Cost Flows in Capacitated Networks with Applications". *The Bell System Technical Journal*, **61**(6), 949-968 (1982).

48. J. Orlin, "Maximum Throughput Dynamic Network Flows". *Math Prog.* **27**, 210-223 (1983).

49. J. Orlin, "Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule: An Application of Periodic Posets". *Operations Research*, 760-776, (1982).

50. C. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*. Prentice-Hall, Englewood Cliffs, N.J. (1982).

51. W. Roege, *Pilot Scheduling in a Fighter Squadron*, MS. Thesis, M.I.T. (1983).

52. R. Russel and W. Igo, "An Assignment Routing Problem". *Networks* **9**(1), 1-17 (1979).

53. L. Schrage, "Formulation and Structure of More Complex/Realistic Routing and Scheduling Problems". *Networks* 11(2), 229-232 (1981).

54. M. Segal, "The Operator Scheduling Problem: A Network Flow Approach". *Ops. Res.* **22**, 808-823 (1974).

55. R. Simpson, *Scheduling and Routing Models for Airline Systems*. Flight Transportation Laboratory, Report FTL-R68-3, M.I.T.,(December 1969).

56. L. Smith, D. Bird and A. Wiggins, "A Computerized System to Schedule Nurses That Recognizes Staff Preferences". *Hospital and Health Services Admin.*, 19-35 (Fall 1979).

57. USAF Air Training Command, *Undergraduate Pilot Training*, Regulation 55-1 Vol 1 (1984).

58. C. Westerberg, B. Bjorklund, and E. Hultman, "An Application of Mixed Integer Programming in a Swedish Steel Mill", *Interfaces* (February 1977).

APPENDIX A

This appendix contains the five daily schedules developed manually by the 559 FTS squadron scheduler. Using contract inputs from each unit the squadron scheduler constructed the complete schedule and assigned tasks to units. Aircraft and cross-country start times are located in the center section of the schedule; simulator start times are in the upper right corner; and SOF/RSU/other start times are in the lower right corner. Using these schedules and the Student preference lists of appendix C the unit schedulers construct daily unit schedules.

```
AILY FLYING PLAN      559 F . Billygoats      Mon 24 eb 86 G |~~~~ SIMS ~~~~|
                                                                  1  2  3  4
501       0845 A     1130 Upg     1415 BLEY    1659 B     1944       0630 D     D
504       0848       1133         1418         1703       1947       0700    D     D
507       0851 F     1136 Upg     1421 D...#   1706 B     1950       0755 C     C
51        0854       1139         1424         1709       1953       0825    E     E
513       0857       1142         1427         1712       1956       0920 A     A
516       0901 F     1145 A       1430 D...#   1715       1959       0950    F     F
519       0904       1148         1433         1718       2003       1045 D     D
522       0907       1151         1436         1721       2006       1115    C     C
525       0910 F     1154 A       1439 E...#   1724       2009       1210 C     E
528       0913       1157         1442         1727       2012       1240    A     A
531       0916 F     1201 C ob    1445 *1630   1730       2015       1335 F     B
534       0919       1204         1448         1733       2018       1405    B     D
537       0922       1207         1451         1736       2021       1500 D     D
540       0925 SE    1210 A       1454 E...#   1739       2024       1530    A     A
543       0928       1213         1457 ^1406   1742       2027       1625 U     U
546       0931       1216         1501         1745       2030       1655    B     B
549       0934 D     1219 C ob    1504 *1648   1748       2033       1750
552       0937       1222         1507 ^1421   1751       2036       1820
555 D     0940 D     1225 F       1510 A...#   1754       2039
558       0943       1228         1513         1757       2042
702       0946       1231         1516 ^1430   1801       2045
705 D     0949 D     1234 A ob    1519 *1704   1804       2048       Sunrise: 0704
708       0952       1237         1522         1807       2051       Sunset : 1828
711       0955       1240         1525 ^1439   1810       2054
714 D     0958 D     1243 F       1528 A...#   1813       2057
717       1002       1246         1531         1816       2101
720       1005       1249         1534         1819       2104
723 C     1008 D     1252 F       1537 F...#   1822       2107
72        1011       1255         1540 ^1454   1825       2110            SOF
7   C     1014 D     1258 F       1543 F..#    1828       2113       0555 Ogle
732       1017       1302         1546 .       1831       2116       1033 Meyer
735       1020       1305         1549         1834       2119       1512 Huse
738 E     1023 F ob  1308 *1452   1552 SE      1837       2122
741       1026       1311         1555         1840       2125       1951 Close
744       1029       1314         1558 ^1510   1843       2128
747 E     1032 D     1317 305xc   1602 *****   1846 ***** 2131 *****      RSU
750       1035       1320         1605         1849       2134       0645 D/D
753 E     1038 C ob  1323 *1508   1608 Upg     1852       2137       0946 E/D
756       1041       1326         1611         1855       2140       1248 Ck/A
759       1044       1329         1614 ^1528   1858       2143       1549 F/A
803 E     1047 E     1332 B       1617 Upg     1902       2146       1851 Close
806       1050       1335         1620         1905       2149
809       1053       1338         1623 ^1537   1908       2152       Dispatcher
812 E     1056 E     1341 B       1626 Dot     1911       2155       0510 Open :
815       1059       1344         1629         1914       2158       1951 Close:
818 Dot   1103 E     1347 B       1632 B       1917       2202
821       1106       1350         1635         1920       2205       Snack Bar
824       1109       1353         1638         1923       2208       0555 C/C
827 A     1112 C ob  1356 *1541   1641 B       1926       2211       1200 B/B
830       1115       1359         1644         1929       2214       1806 Close
833       1118       1403         1647         1932       2217
836 A     1121 Upg   1406 B...#   1650 B       1935       2220
839       1124       1409         1653         1938       2223
842       1127       1412         1656         1941       2410
otal sorties: 82       Cap: 20/3                   *****  PT  *****
          Local   OBL   OBR   XC   NITE   Sims     A: 0512 - 0612
  PIT      42      6     6     0     0     30      B: 1017 - 1117
PGRADE     5       0     0     0     0     2
UPPORT     22      0     0     1     0     0
```

```
AILY FLYING PLAN        559 FT  Billygoats         Tue 25 F[illegible] 86 G |~~~~ SIMS ~~~~|
                                                                         1 2 3 4
501         0845 A     1130 C     1415 D     1659 Uog   1944        0645 D   D
504         0848       1133       1418       1703       1947        0715   E   E
507         0851 A     1136 Uog   1421 D     1706 ATCSE 1950        0810 C   C
51[illegible]         0854       1139       1424       1709       1953        0840   A   A
513         0857       1142       1427       1712       1956        0935 F   F
516         0901 A     1145 Upg   1430 D     1715 ATCSE 1959        1005   D   D
519         0904       1148       1433       1718       2003        1100 B   U
522         0907       1151       1436       1721       2006        1130   C   C
525         0910 SE    1154 C ob  1439 *1624 1724       2009        1225 E   E
528         0913       1157       1442       1727       2012        1255   A   A
531         0916 Chou  1201 E     1445 D     1730       2015        1350 F   F
534         0919       1204       1448       1733       2018        1420   D   D
537         0922       1207       1451       1736       2021        1515 B   C
640         0925 F     1210 C ob  1454 *1639 1739       2024        1545   M   A
543         0928       1213       1457       1742       2027        1640 A   A
546         0931       1216       1501       1745       2030        1710   U   U
549 D       0934 F     1219 E     1504 C     1748       2033        1805 B   B
552         0937       1222       1507       1751       2036        1835   B   B
555 D       0940 F     1225 E     1510 C     1754       2039
558         0943       1228       1513       1757       2042
702         0946       1231       1516       1801       2045
705 D       0949 F     1234 C ob  1519 *1704 1804       2048        Sunrise: 0703
708         0952       1237       1522       1807       2051        Sunset : 1829
711         0955       1240       1525       1810       2054
714 D       0958 D     1243 E     1528 B     1813 Kngy  2057        need IP for Dr.Chou
717         1002       1246       1531       1816       2101
720         1005       1249       1534       1819       2104
723 D       1008 D     1252 A     1537 B     1822 Sch   2107
726         1011       1255       1540       1825       2110             SOF
72[illegible] Rowcl   1014 D     1258 A     1543 B     1828 Dot   2113        0549 E-Collins
732         1017       1302       1546       1831       2116        0927 King
735         1020       1305       1549       1834       2119        1305 Wagner
738 Dot     1023 D     1308 F     1552 B     1837 F.nt  2122        1643 Hamilton
741         1026       1311       1555       1840       2125        2022 Close
744         1029       1314       1558       1843       2128
747 E       1032 D     1317 A ob  1602 *1746 1846       2131             RSU
750         1035       1320       1605       1849       2134        0639 D/D
753 E       1038 D     1323 F     1608 SE    1852       2137        0932 C/D
756         1041       1326       1611       1855       2140        1225 F/A
759         1044       1329       1614       1858       2143        1518 B/A
803 E       1047 B     1332 F ob  1617 *1802 1902       2146        1811 Close
806         1050       1335       1620       1905       2149
809         1053       1338       1623       1908       2152        Dispatcher
812 E       1056 B     1341 F     1626 A     1911       2155        0504 Open :
815         1059       1344       1629       1914       2158        2022 Close:
818 E       1103 B     1347 F ob  1632 *1817 1917       2202
821         1106       1350       1635       1920       2205        Snack Bar
824         1109       1353       1638       1923       2208        0549 C/C
827 C       1112 B     1356 D     1641 Upg   1926       2211        1137 B/B
830         1115       1359       1644       1929       2214        1726 Close
833         1118       1403       1647       1932       2217
836 C       1121 B     1406 D     1650 Upg   1935       2220
839         1124       1409       1653       1938       2223
842         1127       1412       1656       1941       2405
otal sorties: 81        Cap: 20/3
            Local   OBL    OBR    XC    NITE   Sims
    PIT      53      6      6      0     0      32
JPGRADE      5       0      0      0     0      3
SUPPORT      10      0      0      0     1      1
```

```
DAILY FLYING PLAN        559 FTS Billygoats        Wed 26 Feb 86 G |~~~~ SIMS ~~~~|
                                                                           1  2  3  4
0601        0845 E     1130 C       1415 B       1659 3305  1944      0630 D     D
0604        0848       1133         1418         1703       1947      0700    D     D
0607        0851 A     1136 C       1421 B       1706 B     1950      0755 C     C
0610        0854       1139         1424         1709       1953      0825    E     A
0613        0857       1142         1427         1712       1956      0920 F     F
0616        0901 A     1145 C ob    1430 *1615   1715 B     1959      0950    D     D
0619        0904       1148         1433         1718       2003      1045 B     B
0622        0907 SE    1151 DVOrt   1436 B       1721 B     2006      1115    C     C
0625        0910       1154         1439         1724       2009      1210 C     U
0628        0913       1157         1442         1727       2012      1240    E     E
0631        0916 F     1201 Upg     1445 B       1730 B     2015      1335 A     F
0634        0919       1204         1448         1733       2018      1405    F     B
0637        0922 F     1207 Upg     1451 C       1736       2021      1500 C     C
0640        0925       1210         1454         1739       2024      1530    A     A
0643        0928       1213         1457         1742       2027      1625 U     A
0646        0931 F     1216 E       1501 C       1745       2030      1655    B     B
0649        0934       1219         1504         1748       2033      1750
0652        0937       1222         1507         1751       2036      1820
0655 D      0940 F     1225 E       1510 C       1754       2039
0658        0943       1228         1513         1757       2042
0702 D      0946 D     1231 E       1516 C       1801       2045
0705        0949       1234         1519         1804       2048      Sunrise: 0702
0708        0952       1237         1522         1807       2051      Sunset : 1830
0711 D      0955 D     1240 E       1525 C       1810       2054      IP for DVort
0714        0958       1243         1528         1813       2057
0717 D      1002 D     1246 A ob    1531 *1716   1816       2101
0720        1005       1249         1534         1819       2104
0723        1008       1252         1537         1822       2107
0726 Dot    1011 D     1255 A       1540 A.ct    1825       2110           SOF
0729        1014       1258         1543         1828       2113      0602 Mever
0732 ATCSE  1017 D     1302 A       1546 A       1831       2116      1046 Ogle
0735        1020       1305         1549         1834       2119      1530 B-Collins
0738        1023       1308         1552         1837       2122
0741 C      1026 D     1311 F       1555 A       1840       2125      2015 Close
0744        1029       1314         1558         1843       2128
0747        1032       1317         1602         1846       2131           RSU
0750 C      1035 B     1320 F       1605 Chown   1849       2134      0652 C/D
0753        1038       1323         1608         1852       2137      0957 E/D
0756 C      1041 B     1326 F ob    1611 *1755   1855       2140      1303 B/A
0759        1044       1329         1614         1858       2143      1609 F/A
0803        1047       1332         1617         1902       2146      1915 Close
0806 C      1050 B     1335 F       1620 Rowc    1905       2149
0809        1053       1338         1623         1908       2152      Dispatcher
0812 E      1056 B     1341 F ob    1626 *1811   1911       2155      0517 Open :
0815        1059       1344         1629         1914       2158      2015 Close:
0818        1103       1347         1632         1917       2202
0821 E      1106 C     1350 F       1635 Upg     1920       2205      Snack Bar
0824        1109       1353         1638         1923       2208      0602 C/C
0827 E      1112 C     1356 ~~F~~SE 1641 Dot     1926       2211      1216 B/B
0830        1115       1359         1644         1929       2214      1830 Close
0833        1118       1403         1647         1932       2217
0836 E      1121 C ob  1406 *1550   1650 ATCSE   1935       2220
0839        1124       1409         1653         1938       2223
0842        1127       1412         1656         1941       2386
Total sorties: 81         Cap: 21/3                    *****  PT  *****
             Local  OBL    OBR    XC    NITE   Sims   D: 1311 - 1411
    PIT       58     5      5      0      0      30    E: 1525 - 1625
UPGRADE        3     0      0      0      0       2
SUPPORT       10     0      0      0      0       0
```

```
DAILY FLYING PLAN      559 FTS Billygoats      Thu 27 Feb 86 G |~~~~ SIMS ~~~~|
                                                                  1  2  3  4
0601     0845 Dot    1130 C     1415 Upg    1659         1944     0600 D     D
0604     0848        1133       1418        1703         1947     0630 \ D     D
0607     0851 AtcSE  1136 C     1421 Upg    1706         1950     0725 C     C
0610     0854        1139       1424        1709         1953     0755   C     E
0613     0857        1142       1427        1712         1956     0850 E     E
0616     0901 AtcSE  1145 C     1430 A      1715         1959     0920   U     D
0619     0904        1148       1433        1718         2003     1015 D     A
0622     0907        1151       1436        1721         2006     1045   A     C
0625     0910 3305   1154 F ob  1439 *1624  1724         2009     1140 C     C
0628     0913        1157       1442        1727         2012     1210   F     F
0631     0916 Upg    1201 F     1445 A.ct   1730         2015     1305 E     E
0634     0919        1204       1448        1733         2018     1335   B     B
0637     0922        1207       1451        1736         2021     1430 U     A
0640     0925 Upg    1210 F     1454 F      1739         2024     1500   A     F
0643     0928        1213       1457        1742         2027     1555 F     B
0646     0931        1216       1501        1745         2030     1625   B     B
0649     0934 D      1219 E     1504 F      1748         2033     1720 M
0652     0937        1222       1507        1751         2036     1750
0655     0940 D      1225 E     1510 4shio  1754         2039
0658     0943        1228       1513        1757         2042
0702     0946        1231       1516        1801         2045
0705     0949 D      1234 E     1519 4ship  1804         2048     Sunrise: 0701
0708     0952        1237       1522        1807         2051     Sunset : 1831
0711     0955        1240       1525        1810         2054
0714 D   0958 D      1243 E     1528 4shio  1813         2057
0717     1002        1246       1531        1816         2101
0720     1005        1249       1534        1819         2104
0723 D   1008 D      1252 B     1537 4shio  1822         2107
0726     1011        1255       1540        1825         2110           SOF
0729 D   1014 D      1258 B     1543 Chwob  1828 *2013   2113     0614 Watson
0732     1017        1302       1546        1831         2116     1035 Huse
0735     1020        1305       1549        1834         2119     1456 England
0738 D   1023 D      1308 B     1552 Dotob  1837 *2022   2122
0741     1026        1311       1555        1840         2125     1917 Close
0744     1029        1314       1558        1843         2128
0747 C   1032 D      1317 B     1602 B      1846         2131           RSU
0750     1035        1320       1605        1849         2134     0704 D/D
0753 C   1038 SE     1323 SE    1608 B      1852         2137     0950 C/D
0756     1041        1326       1611        1855         2140     1237 A/A
0759     1044        1329       1614        1858         2143     1524 B/A
0803 C   1047 C ob   1332 *1517 1617 B      1902         2146     1811 Close
0806     1050        1335       1620        1905         2149
0809     1053        1338       1623        1908         2152     Disoatcher
0812 E   1056 A      1341 3305  1626 B      1911         2155     0529 Ooen :
0815     1059        1344       1629        1914         2158     1917 Close:
0818 E   1103 A      1347 F ob  1632 *1817  1917         2202
0821     1106        1350       1635        1920         2205     Snack Bar
0824     1109        1353       1638        1923         2208     0614 C/C
0827 E   1112 A ob   1356 *1541 1641        1926         2211     1150 B/B
0830     1115        1359       1644        1929         2214     1726 Close
0833     1118        1403       1647        1932         2217
0836 E   1121 C      1406 Dot   1650        1935         2220     ** Annual soin *
0839     1124        1409       1653        1938         2223     **  seminar   **
0842     1127        1412       1656        1941         2355     * 0800 and 1500
Total sorties: 71      Cao: 20/3                 *****  PT  -*****
           Local  OBL   OBR   XC    NITE  Sims   A: 1730 - 1830
    PIT     42     4     4     0     0     30    B: 0937 - 1037
UPGRADE      4     0     0     0     0      2    C: 1617 - 1717
SUPPORT     13     2     2     0     0      1    D: 1317 - 1417
```

```
DAILY FLYING PLAN      559 F[illegible] Billygoats      Fri 28 [illegible]b 86 G |~~~~ SIMS ~~~~|
                                                                          1  2  3  4
0601         0845 A     1130 E      1415 D      1659        1944        0700 D     D
0604         0848       1133        1418        1703        1947        0730    D     D
0607         0851 Uog   1136 C xc   1421 *****  1706 *****  1950 *****0825 E     E
06[illegible] 0854      1139        1424        1709        1953        0855    E     A
0613         0857       1142        1427        1712        1956        0950 D     B
0616         0901 Uog   1145 E      1430 D      1715        1959        1020    D     D
0619         0904       1148        1433        1718        2003        1115 E     D
0622         0907 B     1151 E      1436 D      1721        2006        1145    S     S
0625         0910       1154        1439        1724        2009        1240 S     B
0628         0913       1157        1442        1727        2012        1310    S     S
0631         0916 B     1201 A xc   1445 *****  1730 *****  2015 *****1405 S     A
0634         0919       1204        1448        1733        2018        1435    D     D
0637         0922 B     1207 B      1451 D      1736        2021        1530 F     C
0640         0925       1210        1454        1739        2024        1600    M     B
0643         0928       1213        1457        1742        2027        1655 B     B
0646         0931 SE    1216 B      1501 D      1745        2030        1725
0649         0934       1219        1504        1748        2033
0652         0937       1222        1507        1751        2036
0655         0940 D     1225 C xc   1510 *****  1754 *****  2039 *****
0658         0943       1228        1513        1757        2042
0702         0946 D     1231 B      1516 F      1801        2045
0705         0949       1234        1519        1804        2048        Sunrise: 0700
0708         0952       1237        1522        1807        2051        Sunset : 1831
0711         0955 F ob  1240 *1425  1525 F      1810        2054
0714         0958       1243        1528        1813        2057
0717 D       1002 D     1246 C xc   1531 *****  1816 *****  2101 *****
0720         1005       1249        1534        1819        2104
0723         1008       1252        1537        1822        2107
1726 D       1011 D     1255 A      1540 A xc   1825 *****  2110 *****       SOF
17[illegible] 1014      1258        1543        1828        2113        0617 King
0732 D       1017 D     1302 C xc   1546 *****  1831 *****  2116 *****1043 Wagner
0735         1020       1305        1549        1834        2119        1509 Smith
0738         1023       1308        1552        1837        2122
0741 D       1026 F ob  1311 *1455  1555 A xc   1840 *****  2125 *****1935 Close
0744         1029       1314        1558        1843        2128
0747         1032       1317        1602        1846        2131             RSU
0750 E       1035 D     1320 C xc   1605 *****  1849 *****  2134 *****0707 D/D
0753         1038       1323        1608        1852        2137        0959 E/D
0756 E       1041 D     1326 Uog    1611 A xc   1855 *****  2140 *****1251 A/A
0759         1044       1329        1614        1858        2143        1543 F/A
0803         1047       1332        1617        1902        2146        1835 Close
0806 E       1050 D     1335 C xc   1620 *****  1905 *****  2149 *****
0809         1053       1338        1623        1908        2152        Dispatcher
0812 E       1056 F     1341 Uog    1626 B      1911        2155        0532 Open :
0815         1059       1344        1629        1914        2158        1935 Close:
0818         1103       1347        1632        1917        2202
0821 ATCSE   1106 F     1350 E.ct   1635 B      1920        2205        Snack Bar
0824         1109       1353        1638        1923        2208        0617 C/C
0827 Dot     1112 E     1356 C xc   1641 *****  1926 *****  2211 *****1203 B/B
0830         1115       1359        1644        1929        2214        1750 Close
0833         1118       1403        1647        1932        2217
0836 3305    1121 E     1406 D      1650 B      1935        2220
0839         1124       1409        1653        1938        2223
0842         1127       1412        1656        1941        2314
Total sorties: 66       Cap: 21/3                      *****  PT  *****
            Local   OBL    OBR    XC    NITE    Sims   C: 1641 - 1741
    PIT       42     2      2     11     0       21    E: 1436 - 1536
 UPGRADE       4     0      0      0     0        2    F: 0610 - 0710
 SUPPORT       5     0      0      0     0        7
```

APPENDIX B

This appendix contains the contracts submitted by the unit schedulers to the squadron scheduler for the week. Each contract requests tasks by type for every day of the week. The requests are based upon instructor availability, duty day limitations, and student training requirements. These contracts are used by the squadron scheduler to determine total task assignments as well as task string assignments to individual units.

WEEKLY SCHEDULING REQUEST

FLIGHT A

WEEK OF 24-28 FEB

X-C REMARKS 4 X-C T-1'S

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTALS
SMITH	1	OFF	1	1	OFF	1	1	STAFF	1	1	OFF	1	OFF	SOF	SOF				
HAVERKAMP	1	1	1	1	1	1	1	O	1	1	O	1	1	X-C					
BURGSTAHLER	1	1	1	1	1	1	1	1	1	1	Box	1		X-C					
ERICHSEN	1	1	1	1	1	1	1	1	1	1	1	1		X-C					
GUEST HELP																			
KENAGY	1			1			1			1			1						
SCHMED		1			1			1			1			1					
IPs AVAILABLE	5	4	4	5	4	4	5	3	4	5	2	4	1	1	0				
TEAM CAPABILITY								1			1								
NUMBER OF TRAINEES	8			8			8			8			8						
X ACFT INCREMENT	.74			.74			.74			.74			.74						
ATTRITION FACTOR	.769			.769			.769			.769			.769						
= AIRCRAFT REQUIREMENT	7.70			7.70			7.70			7.70			7.70						39
X SIM INCREMENT	.44			.45			.44			.45			.44						
= SIMULATOR REQUIREMENT	3.52			3.6			3.52			3.6			3.52						18
T-1 LOCAL AIRCRAFT	~~2~~ 3	~~2~~ 3	~~1~~	~~2~~ 3	~~2~~ 3	~~1~~	2	2	2	2	1		1	1	0				23
T-1 OUT AND BACKS	1			1			1			1									8
T-1 CROSS COUNTRIES															4				16
T-3 AIRCRAFT								1			1								2

T-3 REMARKS T-3 CHECK PREPS FOR CAPT HAVERKAMP

SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED	~~2~~ 3	~~1~~ 2	~~2~~	~~2~~ 3	~~1~~ 2	~~2~~	2	1	2	2	1								18
PHYSICAL TRAINING																			

REMARKS

SORTIES	ACFT	SIMS
REQUESTED	47	20
− REQUIRED	39	18
= ☒ GAIN ☐ LOSS 0.21	8	2
ESTIMATED POSITION	2.15	3.4

SIGNATURE OF FLIGHT COMMANDER

SIGNATURE OF SECTION COMMANDER

WEEKLY SCHEDULING REQUEST

OTO of [illegible] - 10 13 17

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			FLIGHT
Collins	-	- 1	-	T	1	class		mtg	SOF	S		1		1	-				B
Goldsmith	1	1	POP	1	1	1	1	1	class	S	1	PSU	1	1	1				
Bray	1	1	1	1	1	PSU	1	1	1	P	r	1	1	1	1				WEEK OF
Richard	1	1	1	1	1	class	1	1	1	I	1	class	~~1~~	~~1~~	~~1~~				
Cal	1	SM1	1	1	SOS	1	1	1	1		1	1	~~1~~	~~SM~~	~~1~~				
Gray	1	T	1	1	SM	1	1	RSU	1	N	1	1	1	1	1				24 Feb
Willis		1	1	1	1	—	1	1	-	S	1		1	1					X-C REMARKS
	T									S									
										E									
	E									M									
	S									I									
										N									
	T									A									
										R									
GUEST HELP																			
			1			1			1			1			1				
IPs AVAILABLE	-	6	6	6	5	4	6	5	5	-	6	5	~~6~~ 4	~~6~~ 5	~~6~~ 4				
TEAM CAPABILITY			1			1			1			1			1				
NUMBER OF TRAINEES	11			11			11			11			11						
X ACFT INCREMENT	.69			.69			.68			.68			.68						
÷ ATTRITION FACTOR	.769			.769			.769			.769			.769						TOTALS
AIRCRAFT REQUIREMENT	10			10			10			10			9						49
SIM INCREMENT	.45			.46			.46			.46			.46						
= SIMULATOR REQUIREMENT	5			5			5			5			5						25
T-1 LOCAL AIRCRAFT		4	3	4	4	4	4	4	4		4	4	~~4~~ 3	~~4~~ 3	~~3~~ 4				~~52~~ 49
T-1 OUT AND BACKS																			
T-1 CROSS COUNTRIES																			
T-3 AIRCRAFT		^	ORT 2																ORT 2

T-3 REMARKS 1 "POP" sim

SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED		2	2	2	1	1	2	1	2		2	2	~~2~~ 1	~~2~~ 2	~~2~~ 1				20 ~~23~~
PHYSICAL TRAINING																			

REMARKS

Ordering 2 ORTS on Monday

Planning on 5 team sims (10 Trainee's) in the I20xx bit

Request light turn on Monday to give adequate time for Letha seminar and a test first period

Planning on Flying a 'POP' sim 3rd period Monday

SORTIES	ACFT	SIMS
REQUESTED	52	27
− REQUIRED	49 49	25 ~~25~~
= ☑ GAIN □ LOSS	+ ~~3~~	+ ~~4~~
ESTIMATED POSITION	~~1.0~~ 0.7	~~1.0~~ .06

SIGNATURE OF FLIGHT COMMANDER

SIGNATURE OF SECTION COMMANDER

Byron D. Hix

WEEKLY SCHEDULING REQUEST

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY		
KING	0	1	0	1	SoF		1	MTG	0	0	1	0	SoF	0	1			
HARRIS	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
TRUE	1	1	1	1	1	1	RSU	1	1	1	1	1	1	1	1			
LOVELADY	1	1	1	1	RSU		1	1	1	1	RSU		1	1	1			
CURRAN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
LYDA	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
HAY	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
GUEST HELP																		
GUEST + 1'S	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1			
IPs AVAILABLE	6	8	7	7	6	6	6	7	7	6	7	6	6	7	8			
TEAM CAPABILITY	1	1	0	1	0	0	0	2	2	0	1	1	0	0	0			

FLIGHT
86-11
C

WEEK OF
24-28
FEB

X-C REMARKS
(7) 4 HOP
X-C'S
SATURDAY
RETURN

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY	TOTALS
NUMBER OF TRAINEES	13	13	13	13	13		
X ACFT INCREMENT	.77	.77	.77	.77	.76		
+ ATTRITION FACTOR	.769	.769	.769	.769	.768		
AIRCRAFT REQUIREMENT	13.02	13.02	13.02	13.02	12.86		65
SIM INCREMENT	.44	.43	.44	.43	.44		
= SIMULATOR REQUIREMENT	5.72	5.59	5.72	5.59	5.72		29

	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTALS
T-1 LOCAL AIRCRAFT	1	2	1	2	1	1	3	4	5	3	4	0	0	0	0				27
T-1 OUT AND BACKS		4			3			2			1								20
T-1 CROSS COUNTRIES														7					28
T-3 AIRCRAFT																			

T-3 REMARKS

SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED	2	3	2	3	2	2	3	3	2	3	3	0	0	0	1				29
PHYSICAL TRAINING																			

REMARKS

SORTIES	ACFT	SIMS
REQUESTED	75	29
− REQUIRED	65	29
= ☒ GAIN ☐ LOSS	.77	Ø
ESTIMATED POSITION	+1.6	+2.6

SIGNATURE OF FLIGHT COMMANDER

SIGNATURE OF SECTION COMMANDER
Byron D. Huse

WEEKLY SCHEDULING REQUEST

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY		
Meyer	1	1	1	1	1	1	1	MTG	1	1	1	OFF	1	1	1			
Ward	T	D	Y	→														
Embrey	RSU	1	1	RSU	1	1	1	1	1	RSU	1		RSU	1	1			
Merkle	1	1	1	1	1	1	1	1	1	1	1		1	1	1			
Samuels	1	1	1	1	1	1	H	H	G's	1	1		1	1	1			
Heckman	1	1	1	1	1	1	1	1	1	1	1		1	1	1			
Pelnour	1	1	1	1	1	1	1	1	1	1	1		1	1	1			
McMurry	1	1	1	1	1	1	1	1	1	1	1	[illegible]	1	1	1			
GUEST HELP																		
Cliatt		1			1			1			1			1				
Cunningham			1			1			1		1				1			
Duke			1		1			1										

FLIGHT D

WEEK OF 24 Feb

X-C REMARKS

	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTALS
IPs AVAILABLE	6	8	9	6	9	8	6	7	7	6	9	1	6	8	8				
TEAM CAPABILITY	1	1		1		1	1		1	2	1		2	2	1				
NUMBER OF TRAINEES	15			15			15			15			15						
X ACFT INCREMENT	.77			.77			.77			.77			.77						
+ ATTRITION FACTOR	.769			.769			.769			.769			.769						
= AIRCRAFT REQUIREMENT	15			15			15			15			15						75
X SIM INCREMENT	.47			.47			.47			.47			.47						
= SIMULATOR REQUIREMENT	7			7			7			7			7						35
T-1 LOCAL AIRCRAFT	3	7	5	3	7	7	3	5	6	4	8		4	8					70
T-1 OUT AND BACKS																			
T-1 CROSS COUNTRIES																			
T-3 AIRCRAFT			2*																

T-3 REMARKS *ORTS

SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED	4	2	2	4	2	2	4	2	2	4	2		4	2					36
PHYSICAL TRAINING																			

REMARKS

SORTIES	ACFT	SIMS
REQUESTED	70	36
− REQUIRED	75	35
= GAIN LOSS	−0.2	+0.1
ESTIMATED POSITION	+0.3	+1.0

SIGNATURE OF FLIGHT COMMANDER

SIGNATURE OF SECTION COMMANDER

RANDOLPH TW FORM JUN 84 44 (DO) PREVIOUS EDITION IS OBSOLETE. HQ ATC, RANDOLPH AFB TX

24. 25 26 27 28

WEEKLY SCHEDULING REQUEST

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			FLIGHT
Collins				-SoF	—														E
Cannon	1st TNG			—	—	—	—	—	—	—	LV	—	—	—	—	—	—	—	
KARMANN	--	-RSU-	-					-RQ-						-RSU-					WEEK OF 24-28 Feb
O'Brien																			
Richardson																			
Hoagancamp																			
Wartlowski																			X-C REMARKS
GUEST HELP	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	▨	
Lt Col NEGLEY																			
CAPT BERGEZ				1						1			1						
CAPT BRUNING	1						1												
CAPT WILLIK																			
CAPT HUGHES	1	1			1			1			1			1					
IPs AVAILABLE	7	7	7	6	6	6	7	6	6	7	7	6	7	6	6				
TEAM CAPABILITY	—	0	—	—	0	—	—	0	—	—	0	—	—	0	—				
NUMBER OF TRAINEES		8			8			8			8			8					
X ACFT INCREMENT		.68			.68			.68			.69			.68					
ATTRITION FACTOR		.769			.769			.769			.769			.769					TOTALS
= AIRCRAFT REQUIREMENT		7.1			7.1			7.1			7.1			7.1					35.5
X SIM INCREMENT		.47			.47			.47			.47			.47					▨
= SIMULATOR REQUIREMENT		3.76			3.76			3.76			3.76			3.76					18.8
T-1 LOCAL AIRCRAFT	4	3		4	4		4	4		4	4		4	4					41 43*
T-1 OUT AND BACKS																			0
T-1 CROSS COUNTRIES																			0
T-3 AIRCRAFT			2 ORT											1					2 ORT / 1
T-3 REMARKS																			
SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED	3	3		3	3		3	3		3	3		3	2					29
PHYSICAL TRAINING																			

REMARKS

* orientation
AFA

SORTIES	ACFT	SIMS
REQUESTED	41	29
− REQUIRED	35.5	18.8
= ☑ GAIN ☐ LOSS	.8	2.7
ESTIMATED POSITION	+2.3	3.7

SIGNATURE OF FLIGHT COMMANDER

SIGNATURE OF SECTION COMMANDER

WEEKLY SCHEDULING REQUEST

Janet

INSTRUCTOR PILOTS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			FLIGHT
HAMILTON	1	1	1	1	1	SOF	1	1	1		1	1	1	1	1				F
ZIMMERMAN	1	1	1	1	RSU	1	1	1	1		1	1	1	1	1				
RICE	1	1	RSU	1	1	1	1	1	1	SP	1	1	1	1	1				WEEK OF
BOTINE	1	1	1	1	1	1	1	1	1	I	1	1	1	1	1				24-28
REAGAN	ORT			1	1	1	1	1	1	N	1	1	1	1	1				
NICHOLS	1	1	1	1	1	1	1	1	1		1	1	1	1	1				FEB
MILLER	1	1	1	1	1	1	1	1	1	SE	1	1	1	1	1				X-C REMARKS
										M									
										N									
										AR									
GUEST HELP																			
WINKER	1			1				1			1		1		1				
BURKE		1	1		1		1					1		1					
WILLIAM									1		1								
IPs AVAILABLE	7	7	6	8	7	6	9	9	9		9	8	8	7	7				
TEAM CAPABILITY	2	2		2	2					1	3	2	2	2					
NUMBER OF TRAINEES	8			8			7			6			5						
X ACFT INCREMENT	1.11			1.11			1.11			1.11			1.1						
÷ ATTRITION FACTOR	.769			.769			.769			.769			.769						TOTALS
= AIRCRAFT REQUIREMENT	12			11 23			10 33			9 42			7						49
X SIM INCREMENT	.44			.43			.44			.43			.44						
= SIMULATOR REQUIREMENT	4			3			3			3			2						15
T-1 LOCAL AIRCRAFT	4	4		4	4		4	4			2	2		2	2				36
T-1 OUT AND BACKS	1			2			2				2		2						9
T-1 CROSS COUNTRIES																			
T-3 AIRCRAFT			2*			*1													3 LCLs

T-3 REMARKS · *2 ORTS *1 NIGHT TUES

SIMULATORS	MONDAY			TUESDAY			WEDNESDAY			THURSDAY			FRIDAY			SATURDAY			TOTAL
SIMULATORS REQUESTED	2	2		2	2		1	1			2	2							20
PHYSICAL TRAINING																			

REMARKS

SORTIES	ACFT	SIMS
REQUESTED	54	20
− REQUIRED	49	15
= ☑ GAIN □ LOSS	+5	+6
ESTIMATED POSITION	+4.5	+4.0

SIGNATURE OF FLIGHT COMMANDER

Mark W. Barnett

SIGNATURE OF SECTION COMMANDER

APPENDIX C

This appendix contains the student preference lists for units A, B, and C. The lists display each student's name on the right and a listing of instructors to the right. The assigned instructor is circled. The unit scheduler uses the list to determine proper matchings of students the the instructors assigned by the squadron scheduler.

COURSE FV5AAO CLASS 8613 SECTION: 0

STUDENT	INSTRUCTOR NAMES AND FLIGHT CATEGORIES LISTED IN ORDER OF OCCURENCE (LATEST FIRST)						OVER TWELVE
CLARDY WI	HARTMAN S -F	MEYER FRA -IC	EMBREY ST -CI	CLIATT CH -C			
DE LA FEN	MERKLE WI -FCI	BROOKS RO -I	KARMANN D -CF	MEYER FRA -C			
DEDET ROY	MEYER FRA -FCI	MCMURRY J -C	WARD SCOT -IFC				
HANSEN DU	EMBREY ST -C	HOGANCAMP -F	MERKLE WI -C	REDNOUR R -CF	SAMELS GL -CFI	BROOKS RO -I	
HUGENBERG	SAMELS GL -FIC	MCMURRY J -CI	WAGNER RO -I	MERKLE WI -C			
JENSEN DA	SAMELS GL -F	MEYER FRA -C	MCMURRY J -I	WARD SCOT -ICF	CLIATT CH -C		
KIRNAK MI	EMBREY ST -C	REDNOUR R -F	MERKLE WI -ICF	MEYER FRA -I	CLIATT CH -I		
KNOX DAVI	HARTMAN S -CF	LUKE THOM -I	MCMURRY J -FCI	SAMELS GL -F	REDNOUR R -C	WARD SCOT -C	
LEIMER JE	HARTMAN S -FIC	MERKLE WI -C	SHAMBLIN -F				
MOHAMMED	HARTMAN S -ICF	MEYER FRA -C	REDNOUR R -FC	MCMURRY J -C			
MURPHY SA	MEYER FRA -I	EMBREY ST -ICF	REDNOUR R -I	HARTMAN S -F	CLIATT CH -C	CUNNINGHA -C	
OSTROSKI	CUNNINGHA -CIF	EMBREY ST -FI	SAMELS GL -C	WAGNER RO -C			
PRYOR PAU	SAMELS GL -IC	RICHARDSO -F	CLIATT CH -C	HARTMAN S -I			
TANK DANI	REDNOUR R -FIC	HARTMAN S -F	LUKE THOM -C	MEYER FRA -F	SAMELS GL -C	CLIATT CH -F	
WHITNEY R	MERKLE WI -FC	REDNOUR R -FCI	MCMURRY J -I	MEYER FRA -C			

Assigned IP is circled

PCN UE038DL61

(00:11:26)

COURSE FV5AA0 CLASS 8614 SECTION: 0

OVER TWELVE

STUDENT	INSTRUCTOR NAMES AND FLIGHT CATEGORIES LISTED IN ORDER OF OCCURENCE (LATEST FIRST)			
BONDS JAM	[✓ GOLDSMITH -IC]	WILLIS DA -C	CALL WILL -C	
BORN RICH	[✓ GRAY TOM -C]	GILLIS JU -C	BRAY KENN -IC	BEARD MIK -C
CRAVEN RA	GRAY TOM -C	NEILAN MI -C	[✓ CALL WILL -C]	
HOADLEY W	GRAY TOM -I	[✓ WILLIS DA -C]	COLLINS J -C	GOLDSMITH -C
MACDOUGAL	SHAMBLIN -C	COLLINS J -C	[✓ GOLDSMITH -C]	
MONRAD JO	BRAY KENN -C	BEARD MIK -I	[✓ RICHARD J -C]	CALL WILL -C
PARKHURST	[✓ WILLIS DA -IFC]	BRAY KENN -C	WILMOT DA -C	
RANDOLPH	BRAY KENN -C	[✓ CALL WILL -CI]	NEILAN MI -C	
SCHMELTZ	[✓ GRAY TOM -IC]	GOLDSMITH -I	BEARD MIK -C	
SMITH TOB	[✓ RICHARD J -IC]	BEARD MIK -C	CALL WILL -C	
VIGNAROLI	GILLIS JU -C	[✓ BRAY KENN -FIC]		

Assigned IP is Circled

COURSE FV5AAO CLASS 8615 SECTION: 0

STUDENT	INSTRUCTOR NAMES AND FLIGHT CATEGORIES LISTED IN ORDER OF OCCURENCE (LATEST FIRST)			OVER TWELVE
HERLONG C	BERGER ST -C		OBRIEN	
LUNDQUIST	OBRIEN JE -C		CANNON	
MATTISON	KARMANN D -C	BRUNING S -C		
MAZZEI RO	COLLINS J -C	RICHARDSO -C		
PEPPER DA	HOGANCAMP -C			
PRICE JEF	CANNON RO -C		OBRIEN	
REITMEIST	OBRIEN JE -C	CANNON RO -C	WACKOWSKI	
STEWART D	COLLINS J -C	WACKOWSKI -C		

APPENDIX D

This appendix contains the programs used to solve the Level 1 and Level 2 problems. In it we briefly explain how to structure input data, run the programs. Both programs are in Fortran 77 and were compiled and executed on an IBM PC/XT.

D.1 - LEVEL 1 PROGRAM

Both the Level 1 and Level 2 use two input files DATA.MST and SCHED.MST. DATA.MST is the file containing the master data for the programs. It contains the following five items: 1) the total number of tasks in the schedule, 2) the number of instructors available for the day, 3) the number of local aircraft tasks, 4) the number of simulator tasks, and 5) the number of cross-country/out & back tasks. Each entry is a 3 digit number and occupies a separate line. SCHED.MST contains the start times for all tasks in the daily schedule. The order of entry is 1) all local aircraft tasks, 2) simulator tasks, 3) cross country tasks, 4) other tasks. Task start times need not be in order within a block but the block order must be maintained (e.g. all simulator tasks must be in block 2). Entries are in the 4 digit 24 hour clock (e.g. 1315 for 1:15 p.m.) and each entry occupies a separate line. Due to the variable nature of the "other" tasks both the start and end times for these tasks are required. Enter them as an 8 digit number (e.g. 10061421 for a task starting at 10:06 a.m. and ending at 2:21 p.m.). Examples of DATA.MST and

SCHED.MST are included with the Level 1 program. With this input data the Level 1 program will run with the prompt DLWTH15. Output will consist of an infeasibility check and task strings similar to those in figure 5.1. Also, an output file STRTEND is created by the program which contains the start and end times of all tasks in minutes.

data.wed

```
099
035
057
030
005
```

sched.wed

```
0655
0702
0711
0717
0741
0750
0756
0806
0812
0821
0827
0836
0845
0851
0901
0916
0922
0931
0940
0946
0955
1002
1011
1017
1026
1035
1041
1050
1056
1106
1212
1130
1136
1216
1125
1231
1240
1255
1302
1311
1320
1325
1350
1415
1421
```

```
1436
1445
1451
1501
1510
1516
1525
1546
1555
1706
1715
1721
0600
0600
0630
0630
0725
0725
0755
0755
0850
0850
0920
0920
1015
1015
1045
1045
1140
1210
1210
1305
1305
1335
1335
1430
1430
1500
1500
1555
1625
1625
1121
1145
1246
1326
1341
05581046
10311530
15152015
06371000
09451313
12581620
16001915
```

```
C      ***** FINDING THE MINIMUM # OF INSTRUCTORS*****
C
      PROGRAM DILWORTH
      REAL*8 TCOST
      INTEGER C(1800),X(1800),U(1800),RC(1800),B(200)
      INTEGER STARTN(1800),ENDN(1800),CAP(1800)
      INTEGER I1(200),I2(200),FOU(200),NXTOU(1800),FIN(200)
      INTEGER NXTIN(1800),I7(1800),LARGE,TOT,COUNT
      INTEGER STK(1800),STKH(1800),STKM(1800)
      INTEGER ETKH(1800),ETKM(1800),ETK(1800)
      LOGICAL*2 L1(200),L2(200),REPEAT
      COMMON /ARRAYS/STARTN/ARRAYE/ENDN/ARRAYU/U/ARRAYX/X
      COMMON /ARRAY9/RC/ARRAYB/B/BLK1/I1/BLK2/I2/BLK3/FOU
      COMMON /BLK4/NXTOU/BLK5/FIN/BLK6/NXTIN/BLK7/I7
      COMMON /L/N,NA,LARGE
      COMMON /BLK8/L1
      COMMON /BLK9/L2/BLKR/REPEAT
C
C      ***** INPUTTING THE DATA *******
C
      OPEN(5,FILE='DUMP')
      OPEN(33,FILE='DATA.MST')
      READ(33,10)NT
10     FORMAT(I3)
      READ(33,11)NIP
11     FORMAT(I3)
      READ(33,12)NACFT
12     FORMAT(I3)
      READ(33,17)NSIMS
17     FORMAT(I3)
      READ(33,18)NOB
18     FORMAT(I3)
      CLOSE(33)
C
      OPEN(15,FILE='SCHED.MST')
      DO 15 I=1,NACFT
       READ(15,14)STKH(I),STKM(I)
14       FORMAT(2I2)
       STK(I)=STKH(I)*60+STKM(I)-60
       ETK(I)=STK(I)+180
15     CONTINUE
C
      DO 20 I=NACFT+1,NACFT+NSIMS
       READ(15,19)STKH(I),STKM(I)
19       FORMAT(2I2)
       STK(I)=STKH(I)*60+STKM(I)-30
       ETK(I)=STK(I)+135
20     CONTINUE
C
      DO 24 I=NACFT+NSIMS+1,NACFT+NSIMS+NOB
       READ(15,21)STKH(I),STKM(I)
21       FORMAT(2I2)
       STK(I)=0000
       ETK(I)=1400
```

```
24   CONTINUE
C
   DO 23 I=NACFT+NSIMS+NOB+1,NT
    READ(15,22)STKH(I),STKM(I),ETKH(I),ETKM(I)
22     FORMAT(4I2)
    STK(I)=STKH(I)*60+STKM(I)
    ETK(I)=ETKH(I)*60+ETKM(I)
23   CONTINUE
   CLOSE(15)
C
C     ***** OUTPUTTING START AND END TIMES IN MINUTES *****
C
   OPEN(90,FILE='STRTEND')
   DO 27 I=1,NT
    WRITE(90,25) STK(I),ETK(I)
25     FORMAT(2I5)
27   CONTINUE
   CLOSE(90)
C
C     ***** CREATING DUMMY NODES *******
C
   DO 35 I=1,NT
    STK(NT+I)=STK(I)
    ETK(NT+I)=ETK(I)
35   CONTINUE
C
C     ***** GENERATING NETWORK ARCS *****
C
   J=0
   N=2*NT+2
   DO 40 I=1,NT
    DO 45 L=NT+1,2*NT
     IF((STK(L).LT.ETK(I)).OR.(STK(L).GT.ETK(I)+180))GOTO 45
     J=J+1
     STARTN(J)=I
     ENDN(J)=L
     U(J)=1
     C(J)=0
45     CONTINUE
40   CONTINUE
C
C     ***** GENERATING ARCS FROM s AND t *****
C
   DO 50 I=J+1,J+NT
    STARTN(I)=N-1
    ENDN(I)=I-J
    U(I)=1
    C(I)=0
50   CONTINUE
C
   DO 55 I=J+NT+1,J+2*NT
    STARTN(I)=I-J
    ENDN(I)=N
    U(I)=1
```

```
      C(I)=0
55    CONTINUE
C
C      ***** CREATING s-t ARC *****
C
     NA=J+2*NT+1
     STARTN(NA)=N
     ENDN(NA)=N-1
     U(NA)=1000000
     C(NA)=-1
C
C      ***** PLUGGING INTO RELAX  *****
C
     LARGE=20000000
     CALL INIDAT
     REPEAT=.FALSE.
     DO 60 I=1,NA
      RC(I)=C(I)
60    CONTINUE
C
     DO 65 I=1,N
      B(I)=0
65    CONTINUE
C
     CALL RELAX
C
C     ***** CALCULATING NUMBER OF INSTRUCTORS  *****
C
     MIN=NT-X(NA)
     IF(MIN.GT.NIP) THEN
      WRITE(*,67)NIP
67    FORMAT('0','NO FEASIBLE SOL. EXISTS FOR',I4,' INSTRUCTORS')
      WRITE(*,68)MIN
68      FORMAT(' ','THE LOWER BOUND ON THE # OF INSTRUCTORS IS:',I6)
      GO TO 192
     ENDIF
C
C     ***** DETERMINING THE MINIMUM COST ******
C     ***** (COST = TOTAL IDLE TIME) WAY  *****
C     *****  OF USING THE MIN # OF I.P.'S *****
C
     B(N-1)=-MIN
     B(N)=-B(N-1)
     NA=NA-1
99    J=0
C
C     ***** COLLAPSING THE NETWORK  *****
C
     DO 110 I=1,NT
      DO 105 L=1,NT
       IF((STK(L).LT.ETK(I)).OR.(STK(L).GT.ETK(I)+180))GO TO 105
       J=J+1
       STARTN(J)=I
       ENDN(J)=L
      U(J)=1
      C(J)=STK(L)-ETK(I)
      B(I)=0
105     CONTINUE
110   CONTINUE
C
C     ***** MOVING "ARCS TO t" TO COLLAPSED NETWORK *****
C
     DO 120 I=J+NT+1,NA
      STARTN(I)=I-J-NT
      ENDN(I)=N
      U(I)=1
      C(I)=0
120   CONTINUE
C
C     ***** PLUGGING NEW NETWORK INTO RELAX *****
C
     CALL INIDAT
     DO 125 I=1,NA
      RC(I)=C(I)
     CAP(I)=U(I)
125   CONTINUE
     WRITE(*,85)MIN
85    FORMAT('0','THE LOWER BOUND ON THE # OF INSTRUCTORS IS:',I6)
     WRITE(*,100)NT
100   FORMAT('0','THE TOTAL NUMBER OF TASKS SCHEDULED IS:',I5)
     WRITE(*,101)J
101   FORMAT('0','THE TOTAL NUMBER OF ARCS GENERATED IS:',I6)
999   CALL RELAX
     IF(REPEAT) WRITE(*,1000)TOT
1000   FORMAT('0','THE PREVIOUS UPPER BOUND ON COST =',I7)
C
C     ***** CALCULATING THE COST *****
C
     TOT=0
     DO 140 I=1,J
      IF(X(I).NE.1) GO TO 140
      TOT=TOT+C(I)
140   CONTINUE
C
C     ***** OUTPUT *****
C
     WRITE(*,102)
102   FORMAT('0','A POSSIBLE SET OF TASK ASSIGNMENTS IS:')
     COUNT=0
     DO 500 I=1,NT
      DO 600 K=1,J
       IF(X(K).NE.1) GO TO 600
       IF((I.EQ.STARTN(K)).OR.(I.EQ.ENDN(K))) GO TO 500
600     CONTINUE
      WRITE(*,400)I
400     FORMAT(I6)
      COUNT=COUNT+1
500   CONTINUE
```

```
C
C     ***** THIS IS THE HEURISTIC WHICH LIMITS *****
C     ***** INSTRUCTORS TO 3 TASKS OR LESS     *****
C
      NV=0
      DO 800 I=1,J
       IF(X(I).NE.1) GO TO 800
       DO 700 K=1,J
        IF(X(K).NE.1) GO TO 700
        IF(STARTN(K).NE.ENDN(I)) GO TO 700
         WRITE(*,300)STARTN(I),ENDN(I),ENDN(K)
300        FORMAT(3I6)
         COUNT=COUNT+1
C
C      ******** DUTY DAY CHECK ***********
C
         IF(ETK(ENDN(K)).GT.STK(STARTN(I))+720) THEN
          WRITE(*,320)
320         FORMAT(' ','DUTY DAY VIOLATION')
          NV=NV+1
         END IF
C
         X(I)=0
         X(K)=0
          DO 750 M=1,J
           IF(X(M).NE.1) GO TO 750
           IF(STARTN(M).EQ.ENDN(K)) THEN
            WRITE(*,755) ENDN(M)
755          FORMAT(I6)
            COUNT=COUNT+1
            X(M)=0
           ELSE IF(ENDN(M).EQ.STARTN(I)) THEN
            WRITE(*,760) STARTN(M)
760          FORMAT(I6)
            COUNT=COUNT+1
            X(M)=0
           END IF
750         CONTINUE
         GO TO 800
700      CONTINUE
800     CONTINUE
      DO 130 I=1,J
       IF(X(I).NE.1) GO TO 130
       WRITE(*,127) STARTN(I),ENDN(I)
127      FORMAT(2I6)
       COUNT=COUNT+1
C
C      ******* DUTY DAY CHECK **********
C
       IF(ETK(ENDN(I)).GT.STK(STARTN(I))+720) THEN
        NV=NV+1
        WRITE(*,880)
880       FORMAT(' ','DUTY DAY VIOLATION')
       END IF
```

```
C
130  CONTINUE
C
C
C    ******** OUTPUT *************
C
     WRITE(*,888) NV
888    FORMAT('0','THE NUMBER OF DUTY DAY VIOLATIONS =',I4)
C
    L=COUNT+NV
    WRITE(*,890)
890  FORMAT('0','THE SCHEDULE ABOVE IS FEASIBLE')
    WRITE(*,895) L
895  FORMAT(' ','FOR  ',I3,'  INSTRUCTOR PILOTS')
    GO TO 9999
C
    IF((COUNT.EQ.B(N)).AND.(NV.EQ.0)) THEN
     WRITE(*,160)
160    FORMAT('0','THE SCHEDULE ABOVE IS OPTIMAL')
     WRITE(*,170) L
170    FORMAT(' ','FOR ',I3,'  INSTRUCTORS')
     WRITE(*,175)TOT
175    FORMAT(' ','THE OPTIMAL COST =',I7)
     GO TO 192
    END IF
C
C
    IF(L.GT.NIP) THEN
     WRITE(*,70)
70    FORMAT('0','THE SCHEDULE ABOVE IS INFEASIBLE')
     WRITE(*,71) L
71    FORMAT(' ','SINCE IT REQUIRES',I3,'  INSTRUCTORS')
     WRITE(*,72)NIP
72    FORMAT(' ','AND ONLY ',I2,'  ARE AVAILABLE')
     GO TO 192
    END IF
C
C
    IF(L.GT.B(N)) THEN
     WRITE(*,180)
180    FORMAT(' ','THE SCHEDULE ABOVE FEASIBLE')
     WRITE(*,185) L
185    FORMAT(' ','FOR',I3,'  INSTRUCTORS')
     WRITE(*,190)TOT
190    FORMAT(' ','THE UPPER BOUND ON COST(IDLE TIME) =',I7)
    ENDIF
C
C
192    WRITE(*,193)
193    FORMAT('0','DO YOU WISH TO MODIFY THE SCHEDULE?')
194    WRITE(*,195)
195    FORMAT(' ','TYPE IN  1  FOR YES --- 0  FOR NO')
     READ(*,197)ISENS
197    FORMAT(I1)
```

```
       IF(ISENS.EQ.1) THEN
        REPEAT=.TRUE.
199       WRITE(*,200)
200     FORMAT('0','ENTER THE DESIRED # OF INSTRUCTORS IN SCHEDULE')
        WRITE(*,201)MIN
201       FORMAT(' ','THE NUMBER MUST BE GREATER THAN OR EQUAL TO',I4)
        READ(*,202)INEW
202       FORMAT(I2)
        IF(INEW.LT.MIN) GO TO 199
        IF(INEW.GT.NIP) NIP=INEW
        B(N)=INEW
        B(N-1)=-INEW
C
C
        GO TO 99
       ENDIF
       IF(ISENS.NE.0) GO TO 194
9999  CLOSE(5)
      STOP
      END
```

```
THE LOWER BOUND ON THE # OF INSTRUCTORS IS:     38

THE TOTAL NUMBER OF TASKS SCHEDULED IS:    95

THE TOTAL NUMBER OF ARCS GENERATED IS:  1203

THE PREVIOUS UPPER BOUND ON COST =       0

A POSSIBLE SET OF TASK ASSIGNMENTS IS:
   65
   66
   67
   82
   83
   84
   85
   86
   87
   89
   93
    1      23     46
    2      24     47
    3      25     48
    4      26     49
    5      27     50
    6      28     51
    7      29     52
    8      30     53
    9      31     54
   10      32     55
   11      33     70
   12      35     72
   13      36     73
   56
   14      37     74
   15      38     75
   57
   16      39     76
   17      40     77
   58
   18      41     78
   19      42     79
   59
   20      43     80
   88
   21      44     81
   60      34     71
   61      68     95
   64      94     91
   92      22     45
   62      69
   63      90

THE NUMBER OF DUTY DAY VIOLATIONS =    0

THE SCHEDULE ABOVE IS FEASIBLE
FOR     43   INSTRUCTOR PILOTS
Stop - Program terminated.
```

D.2 - LEVEL 2 PROGRAM

In addition to the input files for the Level 1 program the Level 2 program requires two additional input files which are unique to each unit. The first is the unit's Student preference matrix, called ARRAY, which ranks the instructors. The second file is the task assignments for the unit instructors. This file, called TASK, contains the mission number and assigned instructor number for each unit task. Examples of Assign and TASK are included with the Level 2 program. With these two additional input files the Level 2 program will run with the prompt UNIT6. Output will consist of a complete unit schedule including the mission number, instructor number, student number, and a brief description of each unit task as in table 5.6.

```
C       ***** DEVELOPING THE UNIT SCHEDULE *****
C
   PROGRAM UNIT
   REAL*8 TCOST
   CHARACTER*10 NM(20),NMS(40)
   INTEGER C(1600),X(1600),U(1600),RC(1600),B(80),ETKH(200)
   INTEGER STARTN(1600),ENDN(1600),IP(50),T(50),ETKM(200)
   INTEGER I1(80),I2(80),FOU(80),NXTOU(1600),FIN(80),F3(200)
   INTEGER NXTIN(1600),I7(1600),LARGE,TOT,S,R,MSN(50),D5(200)
   INTEGER COST(30,50),ASGN(30,6),ST(200),ET(200),F2(200)
   INTEGER MATE(50,50),ASGT(30,2),A(50),DATA(5),F1(200)
   INTEGER STK(50),ETK(50),D1(200),D2(200),D3(200),D4(200)
   LOGICAL*2 L1(80),L2(80),REPEAT
   COMMON /ARRAYS/STARTN/ARRAYE/ENDN/ARRAYU/U/ARRAYX/X
   COMMON /ARRAY9/RC/ARRAYB/B/BLK1/I1/BLK2/I2/BLK3/FOU
   COMMON /BLK4/NXTOU/BLK5/FIN/BLK6/NXTIN/BLK7/I7
   COMMON /L/N,NA,LARGE
   COMMON /BLK8/L1
   COMMON /BLK9/L2/BLKR/REPEAT
C
C       ***** INPUTTING THE DATA *******
C
   OPEN(60,FILE='DATA.MST')
   DO 8005 I=1,5
    READ(60,8010) DATA(I)
8010   FORMAT(I3)
8005  CONTINUE
   CLOSE(60)
C
   NTOT=DATA(1)
   NACFT=DATA(3)
   NSIMS=DATA(4)
   NOB=DATA(5)
C
   WRITE(*,7777)
7777  FORMAT('0','INPUT THE NUMBER OF STUDENTS TO BE SCHEDULED:')
   READ(*,7778) NSTUD
7778  FORMAT(I2)
   WRITE(*,7779)
7779  FORMAT('0','INPUT THE NUMBER OF UNIT TASKS TO BE SCHEDULED:')
   READ(*,7780) NT
7780  FORMAT(I2)
C
C     ***** INPUTTING THE TASK ARRAY  *****
C
   OPEN(10,FILE='TASK')
   DO 10 I=1,NT
    READ(10,11) MSN(I),IP(I)
11     FORMAT(2I3)
10    CONTINUE
   CLOSE(10)
```

```
      WRITE(*,2000)
2000  FORMAT('0','INSTRUCTOR ASSIGNMENTS FROM DILWORTH RUN:')
      WRITE(*,2005)
2005  FORMAT(' ','TASK #  MISSION #   INSTRUCTOR #')
      DO 1610 I=1,NT
       WRITE(*,1611) I,MSN(I),IP(I)
1611    FORMAT(I4,I10,I15)
1610  CONTINUE
C
C      ******** INPUTTING THE UNIT TASK TIMES ********
C
      OPEN(80,FILE='SCHED.MST')
      WRITE(*,9003)
9003  FORMAT('0','MASTER SCHEDULE TIMES FOR UNIT ACTIVITIES:')
      DO 1 I=1,NTOT
       READ(80,2)D1(I),D2(I),D3(I)
2       FORMAT(I2,I1,I1)
1      CONTINUE
      CLOSE(80)
C
      DO 6 I=1,NT
       F1(I)=D1(MSN(I))
       F2(I)=D2(MSN(I))
       F3(I)=D3(MSN(I))
       WRITE(*,4)F1(I),F2(I),F3(I)
4       FORMAT(I4,I1,I1)
6      CONTINUE
C
C      ***** INPUTTING TASK START AND END TIMES (IN MINUTES)**
C
      OPEN(100,FILE='STRTEND')
      DO 8121 I=1,NTOT
       READ(100,8111) D4(I),D5(I)
8111    FORMAT(2I5)
8121  CONTINUE
      CLOSE(100)
      WRITE(*,6180)
6180  FORMAT('0','START AND END TIMES(IN MINUTES):')
      DO 6300 I=1,NT
       ST(I)=D4(MSN(I))
       ET(I)=D5(MSN(I))
       WRITE(*,6200)ST(I),ET(I)
6200    FORMAT(2I8)
6300  CONTINUE
C
C      ***** INPUTTING THE ASSIGNMENT ARRAY *****
C
      OPEN(20,FILE='ASSIGN')
      DO 20 I=1,NSTUD
       READ(20,12)(ASGN(I,J),J=1,4)
12       FORMAT(4I3)
20     CONTINUE
      CLOSE(20)
      WRITE(*,2001)
```

```
2001  FORMAT('0','STUDENT PREFERENCE MATRIX')
   WRITE(*,2011)
2011  FORMAT(' ','INSTRUCTOR #s')
   DO 1620 I=1,NSTUD
   WRITE(*,1613) (ASGN(I,J),J=1,4)
1613  FORMAT(4I3)
1620  CONTINUE
C
C    *****  CREATING FEASIBILITY MATRIX (MATE)  *****
C
   DO 4000 I=1,NT
    DO 3900 J=1,NT
   IF((ST(J).GE.ET(I)).AND.(ST(J).LE.ET(I)+300))THEN
      MATE(I,J)=301-(ST(J)-ET(I))
     ELSE
      MATE(I,J)=7777
     END IF
3900   CONTINUE
4000  CONTINUE
C
C   *****  WRITE OUT MATE  *****
C
   WRITE(*,4050)
4050  FORMAT('0','MATE MATRIX BASED ON FEAS. OF TASK MATCHINGS')
   DO 4200 I=1,NT
    WRITE(*,4100) (MATE(I,J),J=1,NT)
4100   FORMAT(16I5)
4200  CONTINUE
C
C    *****  INITILIZING COST ARRAY ELEMENTS TO 8888  *****
C
   DO 100 I=1,NSTUD
    DO 50 J=1,NT
     COST(I,J)=8888
50     CONTINUE
100   CONTINUE
C
C    *****  UPDATING COST ARRAY ELEMENTS  *****
C
   DO 1000 I=1,NSTUD
    DO 500 J=1,4
     DO 250 L=1,NT
      IF (ASGN(I,J).EQ.IP(L)) THEN
       COST(I,L)=J
       COUNT=COUNT+1
      END IF
250      CONTINUE
C
500     CONTINUE
C
1000  CONTINUE
C
C     *****  GENERATING ASSIGNMENT ARCS  *****
C
```

```
      R=0
      N=2*NT
1001  J=0
      M=0
      R=R+1
C
      DO 1100 I=1,NSTUD
       DO 1050 K=1,NT
        J=J+1
        STARTN(J)=I
        ENDN(J)=K+NSTUD
        U(J)=1
        C(J)=COST(I,K)
        IF(C(J).LT.5000) M=M+1
1050    CONTINUE
1100  CONTINUE
      IF(M.EQ.0) GO TO 9990
C
C
C      ***** DUMMY NODES AND ARCS *****
C
      DO 1200 I=NSTUD+NT+1,N
       DO 1150 K=1,NT
        J=J+1
        STARTN(J)=I
        ENDN(J)=K+NSTUD
        C(J)=8888
        U(J)=1
1150    CONTINUE
C
1200  CONTINUE
C
      NA=J
C
C     ***** CHECKING # ARCS EQUALS # NODES SQUARED ****
C
      NACHK=NT*NT
      IF(NACHK.NE.NA) THEN
       WRITE(*,101)
101     FORMAT(' ','WE HAVE A PROBLEM BOB')
       WRITE(*,102) NA
102     FORMAT(' ','THE NUMBER OF ARCS IS:',I8)
       WRITE(*,103) NACHK
103     FORMAT(' ','IT SHOULD BE:  ',I8)
       GO TO 9999
      END IF
C
C     ***** NODAL SUPPLY AND DEMAND *****
C
      DO 1300 I=1,NSTUD
       B(I)=-1
1300  CONTINUE
      DO 1400 I=NSTUD+1,NSTUD+NT
       B(I)=1
```

```
1400  CONTINUE
   DO 1500 I=NSTUD+NT+1,N
    B(I)=-1
1500  CONTINUE
C
C    ***** REDUCED COSTS ******
C
   DO 1600 I=1,NA
    RC(I)=C(I)
1600  CONTINUE
C
C     ***** OUTPUTTING DATA *******
C
   WRITE(*,1612)NA
1612  FORMAT('0','NUM ARCS:  ',I8)
   WRITE(*,2002) R
2002  FORMAT('0','COST MATRIX',I4)
   R=R+1
   DO 1630 I=1,NSTUD
     WRITE(*,1627)(COST(I,J),J=1,NT)
1627    FORMAT(16I5)
1630  CONTINUE
C
   LARGE=2000000
   CALL INIDAT
   REPEAT=.FALSE.
C
C
   CALL RELAX
C
    DO 1800 L=1,NSTUD
   DO 1700 I=1,NA
    IF(X(I).NE.1) GO TO 1700
    IF(STARTN(I).EQ.L) THEN
     ASGT(L,1)=STARTN(I)
     ASGT(L,2)=ENDN(I)-NSTUD
    END IF
1700  CONTINUE
1800   CONTINUE
C
C     ***** STORING STUDENT ASSIGNMENTS *****
C
   DO 1810 J=1,NA
    IF((X(J).EQ.1).AND.(C(J).LT.5000)) THEN
     A(ENDN(J)-NSTUD)=STARTN(J)
    END IF
1810  CONTINUE
C
C    *****  RAISING COST OF ASSIGNED TASKS *****
C
   DO 3000 I=1,NSTUD
    K=ASGT(I,2)
    DO 2500 J=1,NSTUD
     COST(J,K)=8888
```

```
2500    CONTINUE
3000  CONTINUE
C
C
    WRITE(*,2900) R
2900  FORMAT('0','COST MATRIX',I4)
    R=R+1
    DO 3100 I=1,NSTUD
     WRITE(*,3050)(COST(I,J),J=1,NT)
3050   FORMAT(16I5)
3100  CONTINUE
C
C     ***** UPDATING THE COST MATRIX BASED ******
C     ***** ON MATE VALUES AND PRIOR ASSIGNMENTS *
C
    DO 6000 I=1,NSTUD
     K=ASGT(I,2)
     DO 5000 J=1,NT
      IF(COST(I,J).GE.5000) GO TO 5000
      IF(MATE(K,J).LT.350) THEN
        COST(I,J)=COST(I,J)*MATE(K,J)
     ELSE
        COST(I,J)=COST(I,J)*MATE(J.K)
      END IF
5000    CONTINUE
C
6000  CONTINUE
C
C     ***** WRITING OUT COST MATRIX UPDATE *****
C
    WRITE(*,6050) R
6050  FORMAT('0','COST MATRIX',I4)
    DO 6100 I=1,NSTUD
     WRITE(*,6075) (COST(I,J),J=1,NT)
6075   FORMAT(16I5)
6100  CONTINUE
C
C     ***** LOOPING FOR NEXT ASSIGNMENT PASS *****
C
    GO TO 1001
C
C     ***** WRITING OUT RESULTS *****
C
    OPEN(200,FILE='INST')
    DO 6900 I=1,20
     READ(200.6910) NM(I)
6910   FORMAT(A10)
6900  CONTINUE
    OPEN(210.FILE='STUD')
    DO 6930 I=1,40
     READ(210.6920) NMS(I)
6920   FORMAT(A10)
6930  CONTINUE
C
```

```
9990 WRITE(*,7000)
7000 FORMAT('1','THE ASSIGNMENTS ARE:')
     WRITE(*,7010)
7010 FORMAT('0','MISSION  INSTRUCTOR #   STUDENT #  TASK TYPE')
     DO 8000 I=1,NT
      IF(MSN(I).LE.NACFT) THEN
      WRITE(*,7050)MSN(I),IP(I),A(I),F1(I),F2(I),F3(I)
7050     FORMAT(' ',I4,I13,I15,I7,I1,I1,' AIRCRAFT')
        GO TO 8000
       ELSE IF(MSN(I).LE.NACFT+NSIMS) THEN
      WRITE(*,7075)MSN(I),IP(I),A(I),F1(I),F2(I),F3(I)
7075      FORMAT(' ',I4,I13,I15,I7,I1,I1,' SIMULATOR')
         GO TO 8000
       ELSE IF(MSN(I).LE.NACFT+NSIMS+NOB) THEN
      WRITE(*,7080)MSN(I),IP(I),A(I),F1(I),F2(I),F3(I)
7080     FORMAT(' ',I4,I13,I15,I7,I1,I1,' CROSS COUNTRY/ O&B')
         GO TO 8000
       ELSE
      WRITE(*,7085)MSN(I),IP(I),A(I),F1(I),F2(I),F3(I)
7085      FORMAT(' ',I4,I13,I15,I7,I1,I1,' SOF/RSU/OTHER')
      END IF
8000 CONTINUE
C
C
9999 STOP
     END
```

END

10— 86

DTIC